

# Toward Establishing an Ontological Structure for the Accumulation of Learning/Instructional Design Knowledge

Yusuke Hayashi<sup>1</sup>, Jacqueline Bourdeau<sup>2</sup>, Riichiro Mizoguchi<sup>1</sup>

<sup>1</sup> ISIR, Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan  
{hayashi, miz}@ei.sanken.osaka-u.ac.jp

<sup>2</sup> LICEF, Télé-université, 100 Sherbrooke W., Montréal, (QC) H2X 3P2 Canada  
jacqueline.bourdeau@licef.ca

**Abstract.** One of the problems to build theory-based authoring systems is the difficulty in building and sustainable maintenance of the knowledge base. This paper discusses an ontological structure for the accumulation of learning/instructional design knowledge. Based on the past results of the OMNIBUS project, we present layering of the scenario model to introduce fundamental conceptualization of WAY knowledge followed by a discussion on how to abstract WAY knowledge from theoretical literature and/or scenarios found in best practices using an example of structuralization of theoretical knowledge.

**Keywords:** Authoring system, Ontology, Accumulation of knowledge

## 1. Introduction

A number of authoring systems have been proposed in the area of learning/instruction support systems [12]. However, even the best of the current authoring systems incorporate a single learning/instructional theory in a procedural way (e.g. [7, 13]) and hence the relationship between the system's behavior and background knowledge derived from a theory is implicit. In short, such a procedural interpretation is conducted by system developers and it is kept only in their brains, which causes the difficulty in sustainable maintenance of a knowledge base according to the modification of knowledge as well as the evolution of theories.

This has been a serious problem not specific to the authoring systems but common to knowledge-based systems in general and is positioned as an issue to be addressed by ontology engineering [2, 3, 10]. In our study, we already proposed a mechanism that consistently enables the declarative definition of theories and their procedural utilization on the basis of the OMNIBUS ontology [11]. Based on the past result on OMNIBUS project<sup>1</sup>, this paper discusses how to accumulate learning/instructional design knowledge, which includes learning/instructional theories and best practices, as the foundation for such theory-awareness.

---

<sup>1</sup> <http://edont.qee.jp/omnibus/>

This paper is organized as follows. Section 2 illustrates the utilization of theoretical knowledge on SMARTIES, which is a prototype of theory-aware and standard-compliant authoring system developed by this study, and the mechanism to realize it from the viewpoint of the knowledge-based system. In order to accumulate learning/instructional design knowledge, Section 3 discusses an ontological structure for the accumulation of learning/instructional design knowledge with the considerations on the categories of strategies stated in learning/instructional theories and the relation between the strategies and the scenario model based on OMNIBUS ontology. Section 4 gives an example to show how learning/instructional design knowledge is accumulated based on the proposed framework. Finally, Section 5 concludes the paper.

## 2. Theory-Aware Support on Authoring Systems

As an example of theory-aware authoring system, this section explains theory-aware support of SMARTIES, which is a prototype of learning/instructional scenario-design support system based on the OMNIBUS ontology. SMARTIES understands theories based on the OMNIBUS ontology and supports the authors to create scenarios that conform to theories. At the same time, this is a standard-compliant system that can output the resultant scenario deliverables in the IMS LD format. Unlike other systems embedded with theories in a procedural manner, the support by SMARTIES is provided based on the declarative knowledge defined by the OMNIBUS ontology, which is developed on Hozo ontology editor<sup>2</sup> and installed outside SMARTIES. Therefore, SMARTIES can automatically change its behavior when OMNIBUS ontology is updated.

### 2.1. Overview of SMARTIES

The current scope of SMARTIES is the design phase, one of the five major phases of instructional design: analysis, design, development, implementation and evaluation. SMARTIES supports the phase from the abstract level to the concrete level, in other words, from goal setting of a scenario to assignment of learning objects (LOs) to it.

In SMARTIES the scenario design is done by decomposing the goal of a scenario into sub-goals using WAY-knowledge [4]. Figure 1 shows a basic unit of the decomposition. One characteristic of this is to model learning/instruction process from two viewpoints: *what* to achieve and *how* to achieve. The oval nodes represent *I\_L events*, in which “I\_L” stands for the relationship between the Instruction and the Learning. An I\_L event describes *what* learner state is achieved by what actions. Black squares linking the macro and the micro I\_L events represent ways of learning/instructional goals achievement (hereafter, WAYs). A WAY describes *how* the state in the macro can be achieved by the sequence of smaller grain-sized states. The other characteristics of this modeling is “OR” relation between WAYs. As shown in Fig. 1, the macro has two WAYs; WAY1 and WAY2. It indicates that there are two alternative ways to achieve the macro (where the learner recognizes what to learn).

---

<sup>2</sup> <http://www.hozo.jp/>

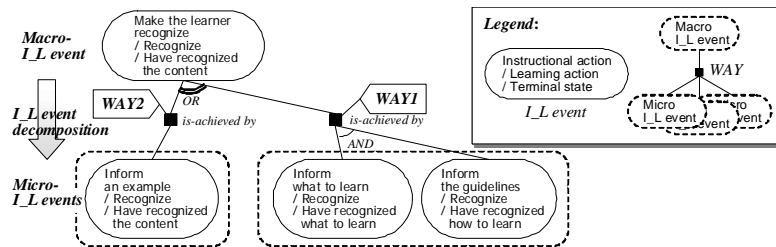


Figure 1. Example of WAY to achieve learning/instructional goals.

This decomposition externalizes the design rationale of the scenario and specifies LOs used in it. Finally, the resultant scenario model is output in the IMS LD format and can be executed on IMS LD compliant tools.

## 2.2. Theory-Aware Support for Learning/Instructional Scenario Design

Figure 2 shows the SMARTIES user interface. The Scenario editor (Fig. 2 (1)) is the main window in which the author makes a scenario model (Fig. 2 (a)) through GUI. In principle, an author can describe a scenario model freely following the I\_L event decomposition scheme shown in Fig. 1. The model can be described by the author in his/her own terms as well as in terms of the concepts defined by the OMNIBUS ontology (Fig. 2 (2)). If the concepts defined by the ontology are used, SMARTIES can interpret the scenario model and offers an intelligent support including explanations of and suggestions on the model based on the theories. For example, the WAY-knowledge proposal window (Fig. 2 (5)) displays the list of applicable WAY-knowledge based on theories (Fig. 2 (d)). This list displays pieces of WAY-knowledge which match with the selected I\_L event in the order of the matching score<sup>3</sup> along the IS-A structure of the theories classified according to the paradigms. When a piece of WAY-knowledge is selected by the author, its structure after application (Fig. 2(e)) and its explanation (Fig. 2(f)) are displayed. All of these contents are created dynamically based on the OMNIBUS ontology. With this information, the author can select one piece of WAY-knowledge that seems to be most relevant.

We have thus far mainly discussed the procedure of scenario model creation by referring to the OMNIBUS ontology. However, if the author cannot find relevant definitions for actions or states in the ontology, the author can also describe an I\_L event in his/her own words, though the support to be provided by the SMARTIES would be limited. In such cases, nevertheless, the design rationale is stored by describing the hierarchical structure of the I\_L events in the scenario model. This is also the same in the WAY description. The scenario author can describe the decomposition of an I\_L event that he/she thinks relevant. In such a case, applicable WAY-knowledge will be provided upon further decomposition as far as the I\_L event is described in terms of the concepts defined by the OMNIBUS ontology.

The validity of a WAY defined by the author can be examined by measuring the degree of its similarity to the WAY-knowledge that SMARTIES has. If the degree of

<sup>3</sup> Matching is done between the I\_L event and the macro I\_L event of the WAY-knowledge.

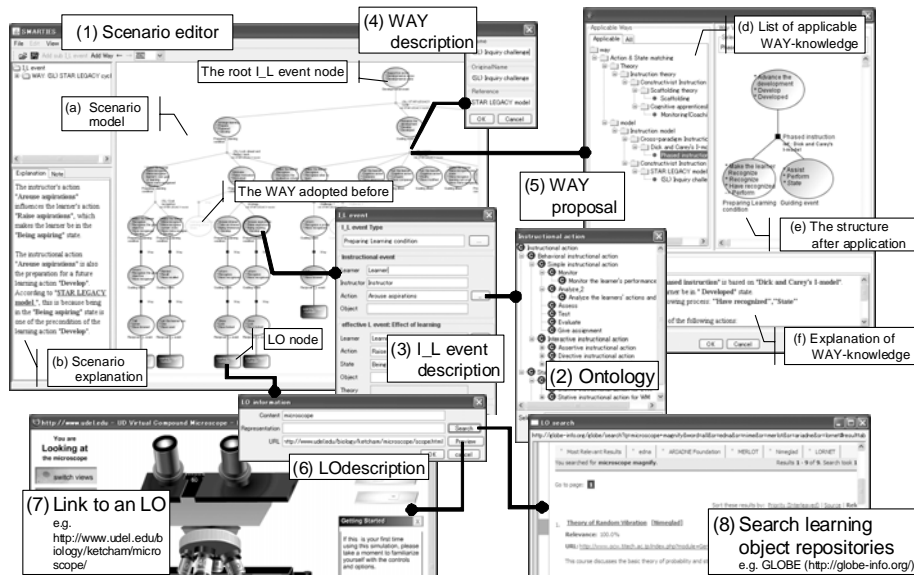


Figure 2. Screenshot of SMARTIES

similarity turns out to be high, we can consider that this WAY defined by the author is likely to be theoretically sound to the extent of the WAY-knowledge that SMARTIES has. The similarity is calculated by comparing the contents of the upper/lower events of the target WAY and the target WAY-knowledge and then normalizing the sum of each similarity. Currently, the similarity is compared by measuring the distance between the concepts regarding the kinds of I\_L events, Instructional action and state for each I\_L event along the IS-A structure of the OMNIBUS ontology. This “distance between the concepts” is calculated using the number of IS-A relations that trace from one concept to the other and the degree of similarity is the inverse of that number. Because this calculation method has plenty of room for improvement, we plan to develop a more sophisticated calculation method.

At the end of the scenario design, learning objects are linked to a scenario model. The leaf nodes depicted by rounded rectangles in the scenario model stand for LOs. In this example, a simulation of a microscope (Fig. 2(6)) is set in order to materialize the leaf I\_L event. This is because the goal in this I\_L event is to remind the learners of the procedure (the manipulation of a microscope) and the transformational media, which are visual images illustrating changes in time or over space, is appropriate to such content according to Clark’s multimedia principle [1]. Authors can set an LO that they made or know as well as search LO repositories for LOs appropriate to the requirement (Fig. 2(7)). Although just the content type is used to discuss the requirement of LO in the current implementation, much more properties are considered to be used to specify the requirement, for example, learner characteristics such as age and prior knowledge, domain characteristics of the content and context characteristics such as mode of instruction and delivery listed in [11]. Enumerating such properties and linking them to LOM elements for LO search is currently in progress.

By these operations, the scenario model is built hierarchically from the abstract level to concrete level as shown in Fig. 2(a). The hierarchical structure indicates the design rationale of the scenario composed of a sequence of the leaves of the scenario model, and its theoretical validity is represented by the pieces of WAY-knowledge used there.

### 2.3. Characteristics of SMARTIES

Because of the functions such as the provision of guidelines or explanation text generation based on the theoretical knowledge, SMARTIES may look like an expert system. However, SMARTIES is fundamentally different from expert systems because its versatile functions are supported simply by the following two basic operations:

1. **A simple read/write operation to/from the ontology:** SMARTIES just needs to know which concepts are defined and what are their “parts” or “attributes” with what constraints, etc.
2. **Pattern matching between I\_L events:** Matching of the pattern of the content of each I\_L event described in the scenario model against the pattern of the upper-level event described in the WAY-knowledge to find the WAY-knowledge applicable to the scenario model.

The following two points enable these two operations to realize the versatile functions described thus far:

- the declarative definition of concepts
- the modeling based on the ontology.

By defining the WAYs as relational concepts in a declarative manner in addition to the basic concept such as actions, states or events, SMARTIES can generate explanations concerning the theories and the content of the scenario model described by the author. By unifying an I\_L event from an upper-level event of a piece of WAY-knowledge and an I\_L event from a scenario model on SMARTIES, the possible decomposition structure of the I\_L event from the scenario model is derived. In other words, the applicability of the theories and their possible results can be indicated<sup>4</sup>. In this manner, SMARTIES processes the support function in a procedural manner based not on heuristic rules but on declarative concept definitions extracted from learning/instructional theories.

However, theoretical knowledge described in a declarative manner is sometimes not sufficient to provide required support. Therefore, using empirical knowledge is also under consideration. For example, SMARTIES can create a learning/instructional process by combining multiple theories within a scenario. Nevertheless, as previously mentioned, the difference in paradigms and theories means a difference in (to a greater or lesser extent) philosophies of learning/instruction. So using too many theories alternately within a scenario may result in a lack of consistency. It is important to

---

<sup>4</sup> This mechanism is not something based on an empirical rule. In other words, this is not something that grasps the changes of the object world by updating the working memory through the WAY-knowledge. This mechanism derives applicable WAY-knowledge by unifying the WAY-knowledge and the scenario model when the author plans learning/instructional processes.

keep consistency to some extent, but the verification of consistency cannot be easily defined by the ontology in a declarative way since there is no theory that supports the criterion for verification. If we deal with such content, use of some heuristic rule to some extent is required. By storing the rules defined by the ontology in a rule base separately from the ontology and using them, we can manage the agreed and well-understood knowledge and ad-hoc, heuristic knowledge separately.

### 3. Toward Accumulation of Learning/Instructional Design Knowledge

We have thus far discussed how WAY-knowledge works as the source of theory-awareness of SMARTIES. In this section, we discuss how such pieces of WAY-knowledge should be accumulated. Although the target of accumulating WAY-knowledge in this study is theoretical knowledge so far, it is also important to abstract practical knowledge from best practices to organize it as useful heuristic rules. Although SMARTIES has a WAY-knowledge editor it is currently just a simple graphical editor yet. In order to accumulate WAY-knowledge from theory or practice, guidelines are required for the abstraction and for organizing knowledge as WAY-knowledge.

#### 3.1. Categorization of Strategies

According to Reigeluth [14] instructional strategies are composed of three different aspects: organizational strategy characteristics, delivery strategy characteristics and management strategy characteristics. Organizational strategy characteristics refer to how instruction will be sequenced, what particular content will be presented, and how this content will be presented. Delivery strategy characteristics deal with what instructional medium will be used and how learners are grouped. Management strategy characteristics include the scheduling and allocation of resources to implement the instruction that is organized and delivered as planned within the previous two strategy aspects [15]. In addition, by Merrill's definition, management strategies involve motivational

Table 1 Categories of strategies.

types	Notes	
	Type of macro state	Type of micro state
Organizational strategy	To specify the sequence of learning/instruction	
Developmental strategy	To specify the development process of learner	
	<i>Internal state</i>	<i>Internal state</i>
Communication strategy	To specify the communication between learner and instructor.	
	<i>Internal state</i>	<i>Communicative state</i>
Component strategy	To specify the content to be presented to learner	
	<i>Communicative state</i>	<i>Communicative state</i>
Delivery strategy	To specify the medium used in the communication	
	<i>Communicative state</i>	<i>Physical state</i>
Management strategy	To specify the cultivation of attitude of learner	
	<i>Internal state</i>	<i>Attitudinal state</i>

techniques, individualization schemes and so on [9]. These categories are considered to be useful as the guidelines for abstracting and organizing knowledge.

According to these categories, this study proposes categories of strategies summarized in Table 1 as the upper level concepts of WAY-knowledge. Basically, these categories are defined according to Reigeluth's definition while Organizational strategy is subdivided into three more detailed sub-categories; Developmental strategy, Communication strategy and Component strategy. The difference among these categories is the target state of decomposition. In other words, the combinations of types of state in the macro and micro I\_L event are different. Using these upper level concepts of WAY-knowledge, this study proposes the structuralization of instructional design knowledge from theory and practice. Actually, those at the upper level are to some extent independent of each theory and WAY-knowledge from each theory or best practice will be placed under the upper level in the IS-A hierarchy. An example of IS-A hierarchy of WAY-knowledge will be taken up in section 4.

### 3.2. Layering a Scenario Model

In this study, as mentioned in 2.1, a learning/instructional scenario is modeled as a hierarchical tree structure composed of the I\_L events from the viewpoint of WAY to achieve learning/instructional goals<sup>5</sup>. In this study, the hierarchical structure is called the "scenario model" whereas only the bottom structure (the sequence of the leaf I\_L events) is called the "scenario." The scenario shows the actual events performed by the learner and the instructor when executed.

This scenario model can be separated into three layers; Rationale layer, Communicative layer, and Presentation layer as shown in Fig. 3. Presentation layer corresponds to a scenario. This layer is linked to LOs and specifies which medium is used for presenting the content to learners. *Physical state* related to the LOs is focused here, for example, *Have looked*, *Have read*, and so on (in this section italic words denote concept defined in the OMNIBUS ontology). Communicative layer is an abstraction of Presentation layer in terms of communication between the learner and the instructor participating in the scenario. This layer deals with *Communicative state*, for example, *Informed*, *Asked* (questions/to do an action) and so on. Rationale layer presents the design rationale of the other two layers. This layer deals with a *Internal state*, for example, *have recognized*, *have recalled* and so on, therefore this can give why the Communicative and Presentation layers are planned in the scenario model from the perspective of the internal change of learners.

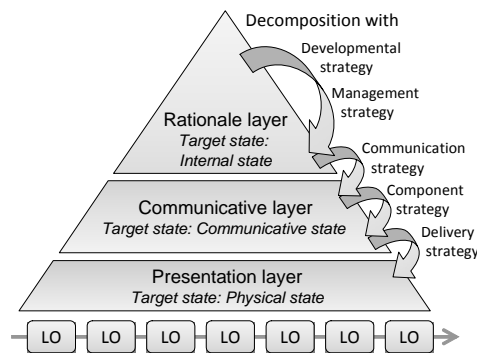


Fig. 3 Layers of a scenario model

<sup>5</sup> This structure is not an "is-a" structure but a "whole and parts" one that is based on the relationship of achievement.

According to the types of states dealt with at each layer, the categories of strategy usable in each layer can be specified. For example, Rationale layer deals with *Internal state*, hence Developmental, Communication and Management strategies can be used to decompose I\_L event in the layer according to Table 1. This would be useful not only for abstracting and categorizing WAY-knowledge from both of theoretical literature or scenarios in best practices but also for characterizing each theory and best practice. To put it little more concretely, each theory and best practice can be characterized by which layer of the scenario model the theory or best practice covers.

#### 4. Example: Structuralization of WAY-knowledge Related to Component Display Theory

We plan to build an upper level concept hierarchy of WAY-knowledge as mentioned above. Based on such hierarchy we aim at organizing WAY-knowledge from abstract level to a certain level for its application to actual learning/instructional scenarios.

Figure 4 shows a part of the IS-A hierarchy of WAY-knowledge including ones based on Component Display Theory (CDT) by Merrill [9]. There are two pieces of upper level WAY-knowledge concepts, which are called *communication strategy* and *component strategy*. These are specialized from left to right in Fig. 4. So, the leftmost WAY-knowledge corresponds to the most general one. The lower level pieces of WAY-knowledge are from CDT in this example. The lowest level pieces of WAY-knowledge can be used for decomposition in a scenario model as shown in Fig. 4(a).

*Communication strategy* deals with how an instructor communicates with a learner. On the other hand, *component strategy* deals with what type of information is provided to a learner by an instructor. The micro I\_L events obtained by the decomposition using *communication strategy* can be further decomposed using the *component strategy* because the micro I\_L event of the former is same as the macro one of the latter (See Fig. 4(b)). This means that an I\_L event decomposed by a subclass of *communication strategy* can be further decomposed by one or some of subclasses of *component strategy*.

The lower level pieces of WAY-knowledge are from CDT. The classification as *Expository* or *Inquisitory* based on the presentation mode by CDT is placed at the lower level *communication strategies*. On the other hand, the classification according to the process of learning/instruction (in CDT the type of process is classified as “presentation”, “practice”, and “performance”) and to the content depending on the objective of the part of the scenario model decomposed by the WAY-knowledge (in CDT the objectives are classified as the following performance level; “remember instance”, “remember generality”, “use”, and “find”<sup>6</sup>) is placed at the lower level *component strategies*. In the CDT these strategies are summarized in a matrix of Primary Presentation Form (PPF) associated with each performance level (see Fig. 7.3 in [9], p. 206). *Component strategy* can also have other types of subclasses, for example, decomposition according to the content such as positive or negative example. Such IS-A hierarchy that is shown in Fig.4 suggests a high possibility to arrange a further

---

<sup>6</sup> These performance level stand for that a learner can do the performance.

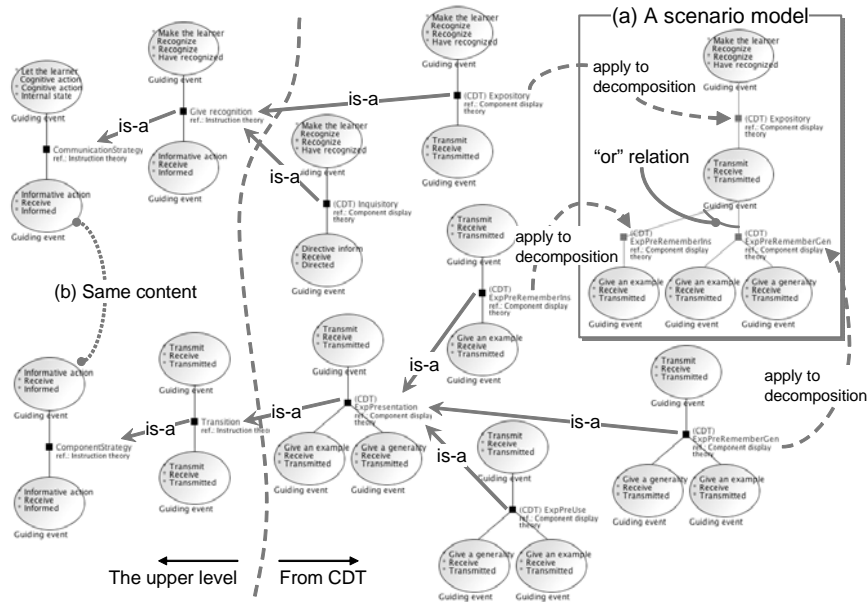


Fig. 4 A part of IS-A hierarchies of WAY-knowledge based on CDT

detailed classification of strategies proposed in CDT from some viewpoints of properties concerned with learning/instructional strategies.

Some pieces of the lowest level WAY-knowledge are used to decompose I\_L event in the sample partial scenario model (Fig. 4(a)). The objective of the top I\_L event is that a learner recognizes the target of learning (the learner is in the state “have recognized”). Firstly the I\_L event is decomposed by *Expository* WAY-knowledge because the I\_L event is same as the macro I\_L event of the WAY-knowledge. Then, the micro I\_L event is further decomposed by either one of the two pieces of WAY-knowledge by CDT because the relation between the two decomposition is OR relation. At the stage of planning these decompositions are just alternatives.

As discussed so far such an IS-A hierarchy of WAY-knowledge can be a guideline for abstracting WAY-knowledge from both of theoretical literature or scenarios in best practices as well as on deciding a piece of WAY-knowledge to decompose an I\_L event.

## 5. Conclusions

This paper discusses an ontological structure for the accumulation of learning/instructional design knowledge. The concept of WAY-knowledge and the organization of it in an IS-A hierarchy proposed by this study can be considered to be helpful in both utilization of learning/instructional design knowledge and abstracting it from theoretical literature and/or scenarios found in best practices. Studies on CSCL script, organization and utilization of patterns are proposed in [4, 7]. These approaches are more practical but less formal than our approach based on the OMNI-

BUS ontology. These two approaches will be complementary to each other and our approach can provide conceptual guidelines for the design of learning processes.

The feasibility of such organization of WAY-knowledge remains as a topic to be discussed further. A further direction of this study will be to provide more evidence for this consideration. OMNIBUS has 99 pieces of WAY-knowledge extracted from 11 theories [6]. We are planning to develop an upper level IS-A hierarchy of WAY-knowledge dependent on each theory and best practice, and to organize such pieces of WAY-knowledge under the upper level IS-A hierarchy of WAY-knowledge. In addition, it calls for further consideration and implementation of support for abstracting learning/instructional design knowledge and placing the result in the IS-A hierarchy of WAY-knowledge.

## References

1. Clark, R.C. and Mayer, R.E.: *e-learning and the science of instruction*, 2nd ed., Preiffer (2007)
2. Dicheva, D.: O4E Wiki, <http://o4e.iiscs.wssu.edu/xwiki/bin/view/Blog/Articles>.
3. Devedzic, V.: *Semantic Web and Education*, Springer (2006)
4. Harrer, A.: "An Approach to Organize Re-usability of Learning Designs and Collaboration Scripts of Various Granularities". *Proc. of ICALT 2006*, pp. 164-168 (2006)
5. Hayashi, Y., Bourdeau, J. and Mizoguchi, R. : "Ontological Support for a Theory-Eclectic Approach to Instructional and Learning Design", *Proc. of EC-TEL2006*, 155-169 (2006)
6. Hayashi, Y., Mizoguchi, R., and Bourdeau, J.: "Structurization of Learning/Instructional Design Knowledge for Theory-aware Authoring systems", *Proc. of ITS2008*, to appear (2008) (<http://www.ei.sanken.osaka-u.ac.jp/pub/hayashi/ITS2008-hay-CR.pdf>)
7. Hernandez, D., Villasclaras, E.D., Asensio, J.I., Dimitriadis, Y.A., Retalis, S.: CSDL Scripting Patterns: Hierarchical Relationships and Applicability. *Proc. of ICALT 2006*, pp. 388-392 (2006)
8. Koedinger, K. R., Alevan, V. A. W. M. M., & Heffernan, N. T.: Toward a Rapid Development Environment for Cognitive Tutors. *Proc. of AIED 2003*, pp. 455-457 (2003)
9. Merrill, M.D.: "Component display theory", *Instructional-design theories and models: An overview of their current status*. Hillsdale, pp. 279-333 (1983)
10. Mizoguchi, R. and Bourdeau, J.: "Using Ontological Engineering to Overcome Common AI-ED Problems", *International Journal of Artificial Intelligence in Education*, Vol.11, No.2, pp. 107-121 (2000)
11. Mizoguchi, R., Hayashi, Y., and Bourdeau, J.: "Inside Theory-Aware and Standards-Compliant Authoring System", *Proc. of SWEL'07: Ontologies and Semantic Web Services for Intelligent Distributed Educational Systems*, pp. 1-18 (2007)
12. Murray, T., Blessing, S., Ainsworth, S.: *Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive and Intelligent Educational Software*, Springer (2003)
13. Nkambou, R., Gauthier, G., Frasson, C.: CREAM-Tools: An Authoring Environment for Curriculum and Course Building in an Intelligent Tutoring System, *Proc. of CALICSE'96*, p.186-194 (1996)
14. Reigeluth, C. M.: *Instructional Design: What Is It and Why Is It?*, *Instructional-design theories and models: An overview of their current status*. Lawrence Erlbaum Associates, Inc., pp. 3-36 (1983)
15. Smith P. L., Ragan, T. J.: "A Framework for Instructional Strategy Design", *Instructional Design 3<sup>rd</sup> Ed.*, John Wiley & Sons, Inc., pp. 127-150 (2005)