EASE: Evolutional Authoring Support Environment

Lora Aroyo[1], Akiko Inaba[2], Larisa Soldatova[2], and Riichiro Mizoguchi[2]

[1]Department of Computer Science and Mathematics
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
l.m.aroyo@tue.nl

[2]ISIR, Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
{ina,larisa,miz }@ei.sanken.osaka-u.ac.jp

Abstract. How smart should we be in order to cope with the complex authoring process of smart courseware? Lately this question gains more attention with attempts to simplify the process and efforts to define authoring systems and tools to support it. The goal of this paper is to specify an *evolutional perspective* on the Intelligent Educational Systems (IES) authoring and in this context to define the authoring framework EASE: powerful in its functionality, generic in its support of instructional strategies and user-friendly in its interaction with the author. The *evolutional authoring support* is enabled by an authoring task ontology that at a meta-level defines and controls the configuration and tuning of an authoring tool for a specific authoring process. In this way we achieve more control over the evolution of the intelligence in IES and reach a computational formalization of IES engineering.

1. Introduction and Background

For many years now, various types of Intelligent Educational Systems (IES) have proven to be well accepted and have gained a prominent place in the field of courseware [15]. IES also have proven [8, 14] that they are rather difficult to build and maintain, which became, and still is, a prime obstacle for their wide spread popularization. The dynamic user demands in many aspects of software production are influencing research in the field of intelligent educational software as well [1]. Problems are related to keeping up with the constant requirements for flexibility and adaptability of content and for reusability and sharing of learning objects [10].

Thus, the IES engineering is a complex process, which could benefit from a systematic approach, based on a common models and a specification framework. This will offer a common framework, to identify general design and development phases, to modularize the system components, to separate the modeling of various types of knowledge, to define interoperability points with other applications, to reuse subject domains, tutoring and application independent knowledge structures, and finally to achieve more flexibility and consistency within the entire authoring process. Beyond the point of creation of IES, such a common engineering framework will allow for structured analysis and comparison of IES and their easy maintainability.

Currently, a lot of effort is focused on improving of IES authoring tools to simplify the process and allow time-efficient creation of IES [14, 17, 21]. Despite this massive effort, there is still no complete integrated methodology that allows to distinguish between the various stages of IES design, and also to (semi-)automate the modeling and engineering of IES components, as well as providing structured guidance and feedback to the author. There are efforts to decrease the level of complexity of ITS building by narrowing down the focus to a set of programming tasks and tools to support them [5], and by limiting the view to only correct or incorrect 'solutions to a set of tasks' [18]. As a way to overcome the complexity without decreasing the level of 'intelligence' in IES, [18] proposes an approach for separation of authoring components, and [14] offers a KBT-MM a reference model for authoring system of a knowledge-based tutor, which is storing the domain and tutoring knowledge in "modular components that can be combined, visualized and edited in the process of tutor creation".

A considerable amount of the research on knowledge-based and intelligent systems moves towards concepts and ontologies [13] and focuses on knowledge sharing and reusability [9, 11]. Ontologies allow the definition of an infrastructure for integrating IES at the knowledge level, independent of particular implementations, thus enabling knowledge sharing [7]. Ontologies can be used as a basis for development of libraries of shareable and reusable knowledge modules [2] and help IES authoring tools to move towards semantics-aware environments.

In compliance with the principles given by [14] we present an integrated framework that allows for a structured approach to IES authoring, as well as for automation of authoring activities. Characteristic aspect of our approach is the definition of different ontology-based IES intelligence components and the definition of their interaction. We finally aim in obtaining an *evolutional (self-evolving) authoring system*, which will be able to reason over its own behavior and subsequently change it if is necessary. In Section 2 we illustrate aspects of the authoring support process. In Section 3 we consider IES in terms of a reference model. In Section 4 we describe the EASE framework for IES authoring, and subsequently in Section 5 we describe an EASE-based architecture.

2 Authoring Support Approach

The approach we take follows up on the efforts to elicit requirements for IES authoring, define a reference model and modularize the architecture of IES authoring tools. We describe a model-driven design and specification framework that provides functionality to bridge the gap between the author and the authoring system by managing the *increased intelligence*. It accentuates the separation of concerns between subject domain, user aspects, application and the final presentation of the educational content. It allows to overcome inconsistencies and to automate the authoring tasks. We show how the scheme from [14] can be filled with the 'entire intelligence of IES', split into collaborative knowledge components.

First, we look at the increased intelligence. Authoring of IES is a process with an exponentially growing complexity and it requires many different types of knowledge and considering various constraints, requirements and educational strategies [16]. Aiming at (semi)-automated IES authoring we need to have explicit representations of the strategic knowledge (rules, requirements, constraints) in order to be able to reason within different authoring contexts and situations. Managing of the increased intelligence is therefore a key issue in authoring support.

Second, we consider the conceptual distance between the user and the system. According to [13, 17] the authoring tools are neither intelligent nor user-friendly. Special-purpose systems provide extensive guidance, but the disadvantage is that changing such systems is not easy, and the knowledge and content can hardly be reused for their educational purposes [15]. Thus, structured guidance is needed in this complex authoring process.

Our ultimate aim is to attain seemingly conflicting goals: to define authoring support in a powerful, generic and easy to use way. The *power* comes from the use of ontology-based approach. The *generality* is achieved with the help of a meta-authoring tool, instantiated with the concrete learning context to achieve also the power of a domain specific tool. The *ease of use* comes from the combination of the previous two. A characteristic aspect of our approach is the use of Authoring Task Ontology (ATO) [3] as part of the authoring environment, which enables us to build a meta-authoring tool [4] and to tailor the general architecture to the needs of each individual system.

3. Intelligent Educational Systems

Characteristically, ITS [14], maintain and work with knowledge of the *expert*, *learner*, and *tutoring strategies*, to capture the student's understanding of the domain and to tailor instructional strategies to the concrete student's needs. Adaptive Hypermedia reference architectures [8] define a *domain*, a *user* and an *adaptation (teaching) model* used to achieve the content adaptation.

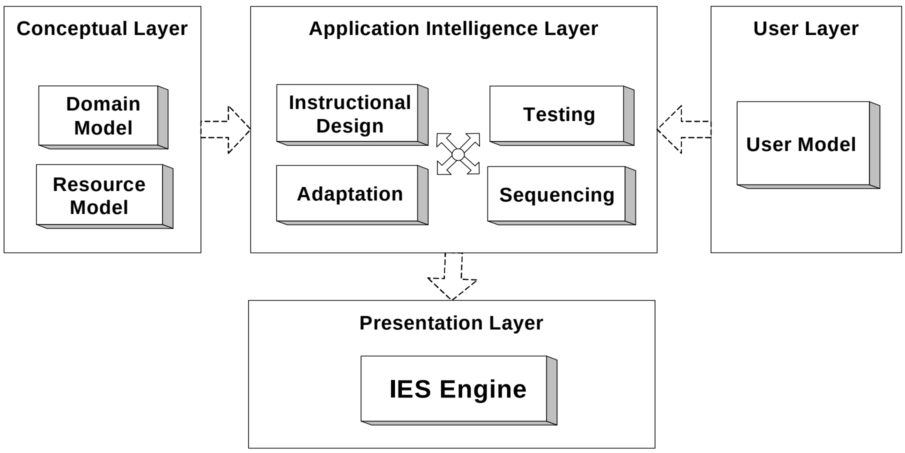

Fig. 1. IES Reference Model

Analogously, Web-based Educational Systems [2] distinguish a *domain*, a *user* and an *application models*, connecting the domain and user models to give a personalized view of the learning resources. A *task model* specifies the concrete sequence of tasks in an adaptive way. As a consequence, [4] distinguish three IES design stages: (1) conceptual modeling of domain and resources, (2) the modeling of application aspects, and (3) simulated use of the user model. Thus, the provision of user-oriented (adapted) instruction and adequate guidance in IES depends on:

- maintaining a model of the *domain*, describing the structure of the information content within IES (based on concepts and their relationships);
- maintaining a personalized portal to a large collection of well organized and structured learning/teaching material *resources*.
- maintaining a model of the *user* to reflect the user's preferences, knowledge, goals, and other relevant instructional aspects;
- maintaining the application intelligence in *instructional design*, *testing*, *adaptation* and *sequencing models*;
- a specific engine to execute the prepared educational structure or sequences.

We organize the common aspects of IES in a model-driven reference approach to allow for a modularization of authoring concerns interoperability of IES components.

4. IES Authoring Context

In line with the IES model defined in the previous section we structure the complexity of the entire authoring process by grouping various authoring activities to:

- model the *domain* as a representation of the domain knowledge;
- annotate, maintain, update and create *learning objects*;
- define the *learning goals*;
- select and apply *instructional strategies* for individual and group learning;
- select and apply *assessment strategies* for individual and group learning;
- specify a *learner model* with learner characteristics;
- specify *learning sequence(s)* out of learning and assessment activities.

To support these authoring tasks we employ knowledge models and capture all the processes related to those tasks in corresponding *authoring modules* as shown in Figure 2. It defines three levels of abstraction for building an IES. At the *product level* we see the final IES. At the *authoring instance level* the actual IES authoring takes place by instantiation of the meta-schema with the actual IES authoring concepts, models and behavior. At the *meta-authoring* we exploit the generic authoring task ontology (ATO) [3, 4] as a main knowledge component in a meta-authoring system and as a conceptual structure of the entire authoring process. A repository of domain-independent authoring components is defined at this level.

At the instance level we exploit ontologies as a way to conceptualize the authoring knowledge in IES. Corresponding ontologies (e.g. for Domain Model, Instructional Strategies, Learning Goal, Test Generation, Resource Management, User Model) are defined to represent the knowledge and important concepts in each of those authoring modules.

Our final goal with this three-layer approach is to realize an evolutional (self-evolving) authoring system, which will be able to reason over its own behavior and based on statistical and other intelligent computations will be able to add new rules or change existing ones in the different parts of the authoring process.

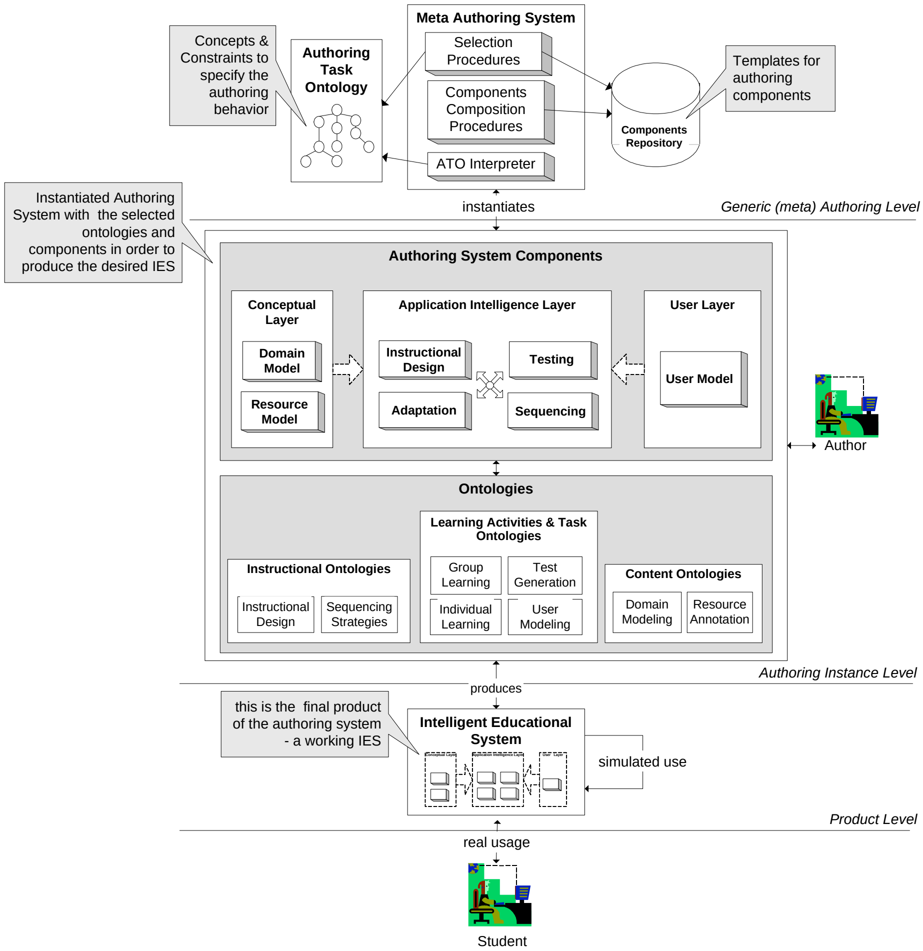

Fig. 2. The IES Authoring Process as captured further in EASE

5. EASE Architectural Issues

To achieve separation of data (content), application (educational strategy), the instructional goals and the assessment activities, we take a goal-centered approach, where a learning goal module is separated from the knowledge on instructional strategies and course sequencing. This allows high reusability of general knowledge on instructional design and strategies. Thus, we have a clear distinction between the content and the computational knowledge, where the learning goal plays a connecting role in order to bring them together within the specific context of each IES.

For example, in Figure 3, the *Collaborative Learning Strategy (CLS)* authoring module provides appropriate group learning strategies for intended users, and requirements for the strategies to the author via the *Sequence Strategies Authoring (SS)* module. To generate explanations and guidance about the recommended strategies *CLS* uses Collaborative Learning Ontology which is a system of concepts to represent collaborative learning sessions and Collaborative Learning Models inspired by learning theories [12, 20].

Another example is given by the *Assessment (A)* module which provides assistance to the author in assessing the learner's (or group of learners) level of understanding and in checking whether a learning goal has been achieved. It uses a test ontology [19] to estimate the effectiveness of the learning process and the preparation/selection of learning objects.

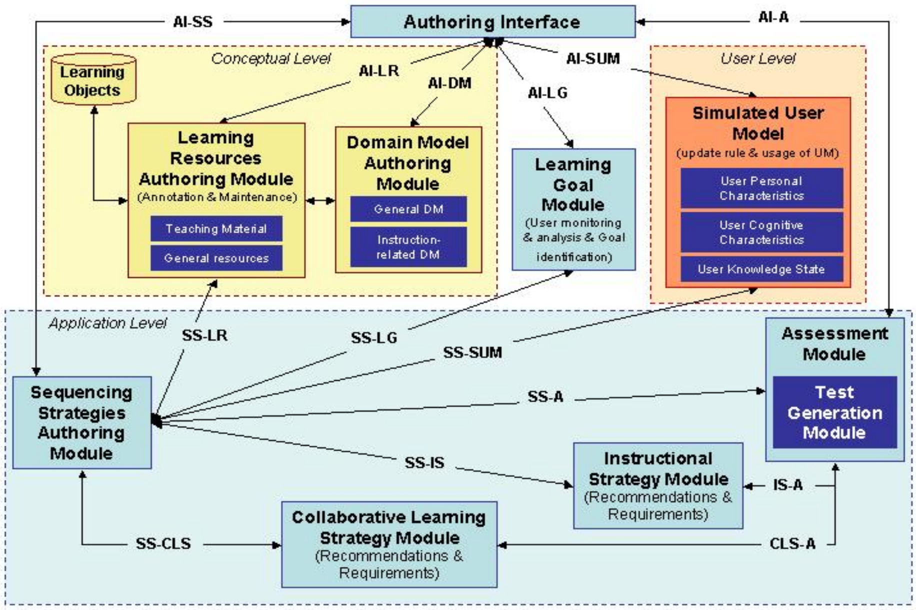

Fig. 3. EASE Reference Architecture

In EASE we follow explicitly the principles supported also by KBT-MM [14] to separate 'what to teach' into modular units independent of 'how to teach' and to present learning goals separately from the instructional content. The rest of the principles we follow implicitly with our use of ontology-based models.

5.1 Communication

The core of the intelligence in the EASE architecture comes from the communication or interactions between the components. There are two "central" components here, the *Sequencing Strategies Authoring (SS)* and the *Authoring Interface (AI)*. The *AI* is the access point for the author to interact with the underlying concepts, models and content. The *SS* interacts with the other components in order to achieve the most appropriate learning sequence for the targeted learner. In this section we illustrate the communication exchange among EASE components, which will further result in the authoring support guidance provided by an EASE-based authoring system.

5.2.1 Authoring Interface (AI) Interactions

At a *conceptual level* the IES author interacts with the *Learning Resources (LR)* and with the *Domain Model (DM)* authoring modules, for example to handle the learning objects. While the author is working with *DM*, an interaction is required between *DM* and *LR* to determine available resources to link to domain concepts. At the *user (learner) level* the author interacts with the *Simulated User Model (SUM)* component in order to determine the use of *UM* (update rules) within the IES application. At the *application level* the author interacts with the *A* and *SS* modules.

5.2.2 Sequencing Strategies (SS) Interactions

In order to realize the most suitable learning task sequence for individual learners, *SS* interacts with *LR, LG, SUM, A, IS* and *CLS* to estimate learner's current knowledge, cognitive state and learning phase. A main role here plays the interaction with *SUM* to adjust the sequencing to the relevant attributes and their values in the user model. *SS* consults *A* for the right evaluation of the user's states and *A* consults SS about the learning history, knowledge values of domain concepts, cognitive states and assessment goals. The *SS* interactions with *A* via *CLS* are presented in Table 1.

Table 1. Sequencing Strategies interactions with Assessment via *CLS*

SS-CLS	CLS-SS
`SendLOCharacteristics`	`RequestAll`
`SendAttrValues`	`SendLearningStyle`
`SendCognitiveAttrValues`	`SendGLstrategy`
`SendLearningHistory`	`ReqsGroupFormation`
`SendLearningGoals`	`ReqsInteractionTypes`
	`ReqsGLsequenceTypes`
	`ReqsGLproblemTypes`
	`ReqsGLtools`
	`SelectLearningGoal`

CLS-A	A-CLS
`SelectFromGoalList`	`ListTestGoals`
`Fill-InTestProperties`	`ListTestPropoerties`

5.3 Example of IES Authoring Interactions

In order to illustrate in practice the intelligence of the IES authoring architecture we will look at the interactions of the *Assessment (A)* authoring module. A typical example is given in Figure 4: an author wants to make a test to assess the learners knowledge after studying a theme. For this, *A* infers an assessment goal, test properties, learner's and domain characteristics from the interaction with *SS* and *IS*. Further, *A* provides an explanation of the most important actions. *A* generates test items and allows the author to edit them, then checks their compatibility with the domain and the test structure. The output of *A* to the author is a generated test, the test documentation, recommendations how to improve the test if necessary, and test characteristics. After the test application *A* interprets the results and checks whether they correspond to the teaching goal.

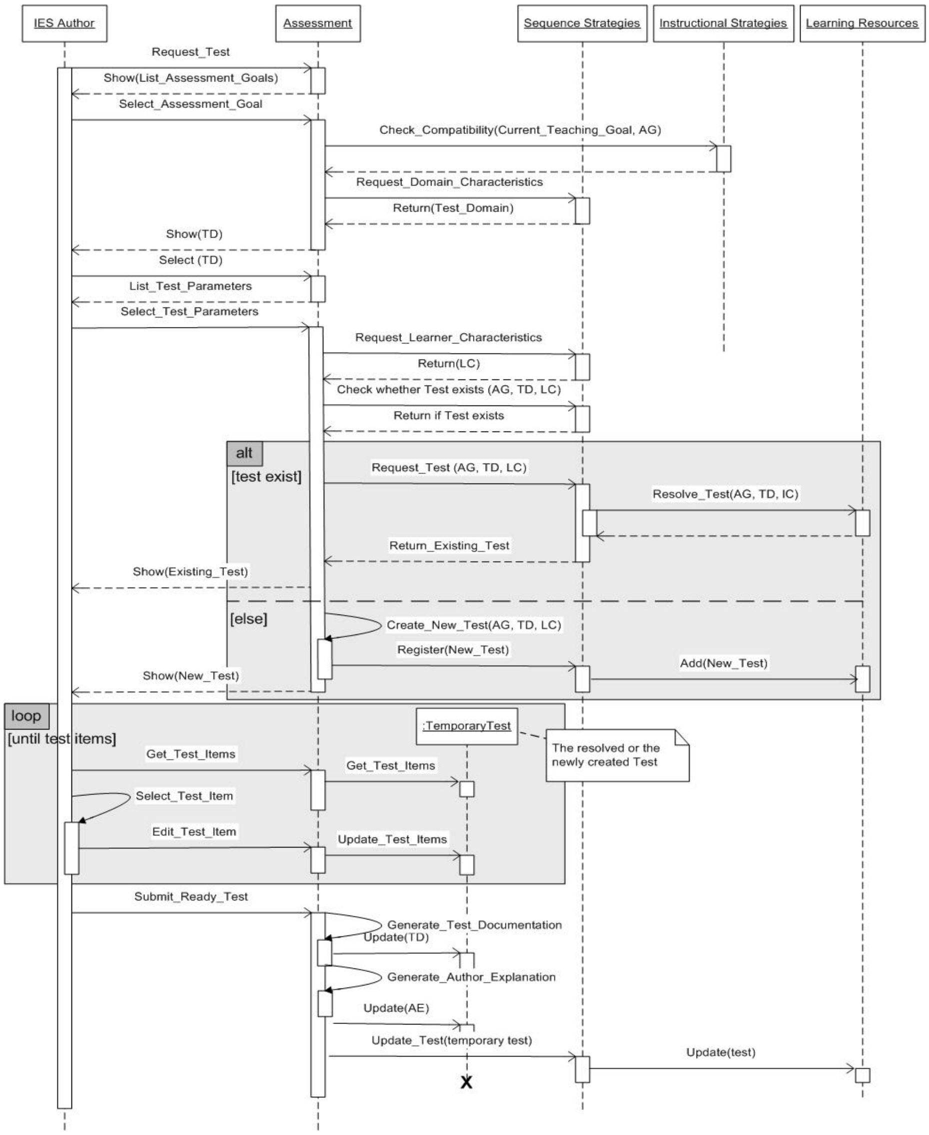

Fig. 4. Assessment Module Interactions

Authoring rules in the *Assessment* knowledge base trigger interaction in order to realize various aspects of the test generation process. For example:

```
IF   author: Edit a test item TI
THEN system: Update the test item
AND  system: Check if TI compatible with the test domain
AND  system: Alert if not compatible with test domain
AND  system: Check if TI compatible with the test structure
AND  system: Alert if not compatible with test structure
```

An authoring support rule in the *CLS's* knowledge base on the other hand produces recommendations and can be triggered by either the author or the system. For example:

```
IF    system:  Identify many appropriate GL strategies
THEN  system:  Show a list of appropriate GL strategies
AND   system:  Explain why each strategy is appropriate
AND   system:  Request to choose one from the list
```

6. Conclusion

Our aim in this research is to specify a general authoring framework for content and knowledge engineering for Intelligent Educational Systems (IES). The main added value of this approach is that on the one hand the ontologies in it make the authoring knowledge explicit, which improves the basis for sharing and reusing. On the other hand, it is configurable through an evolutional approach. Finally, this knowledge is implementable, since all higher-level (meta-level) constructs are expressed with a limited class of generic primitives out of lower-level constructs. Thus, we set the ground for a new generation of evolutional authoring systems, which meet the high requirements for flexibility, user-friendliness and efficiency in maintainability.

We have described reference model for IES and in connection with it a three-level model for IES authoring. For this EASE framework we have identified the main intelligence components and have illustrated their interaction. Characteristic for EASE is the use of ontologies to provide common vocabulary and common understanding of the entire IES authoring processes. This allows for interoperation between different applications and authors.

Acknowledgements

The work is supported by the Mizoguchi Lab, Osaka University, Japan. Special thanks to Prof. Mitsuru Ikeda for his comments on the ATO idea.

Reference

1. Ainsworth, S., Major, N., Grimshaw, S., Hayes, M., Underwood, J., Williams, B., & Wood, D. (2003). REDEEM: Simple Intelligent Tutoring Systems from Usable Tools, In Murray, Ainsworth, & Blessing (eds.), Authoring Tools for Adv. Tech. Learning Env., 205-232.
2. Aroyo, L., Dicheva, D., & Cristea, A. (2002). Ontological Support for Web Courseware Authoring. In Proceedings of ITS 2002 Conference, 270-280.
3. Aroyo, L, & Mizoguchi, R. (2003). Authoring Support Framework for Intelligent Educational Systems. In Proceedings of AIED 2003 Conference.
4. Aroyo, L. & Mizoguchi, R. (2004). Towards Evolutional Authoring Support. Journal for Interactive Learning Research. (in print)
5. Anderson, J., Corbett, A. Koedinger, K., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. The Journal of the Learning Sciences, 4(2), 167-207.

6. Bourdeau, J., Mizoguchi, R. (2000). Collaborative Ontological Engineering of Instructional Design Knowledge for an ITS Authoring Environment, In Proceedings of ITS 2000 Conference.
7. Breuker, J., Bredeweg, B. (1999). Ontological Modelling for Designing Educational Systems, In Proceedings of the Workshop on Ontologies for Intelligent Educational Systems at AIED'99 Conference.
8. Brusilovsky, P. (2003). Developing Adaptive Educational Hypermedia Systems. In Murray, Ainsworth, & Blessing (eds.), Authoring Tools for Advanced Technology Learning Environments, Kluwer Academic Publishers, 377-409.
9. Chen, W. Hayashi, Y., Kin, L. Ikeda, M. and Mizoguchi, R. (1998) Ontological Issues in an Intelligent Authoring Tool, In Proceedings of ICCE 1998 Conference, (1), 41-50.
10. Devedzic, V., Jerinic, L., Radovic, D. (2000). The GET-BITS Model of Intelligent Tutoring Systems. Journal of Interactive Learning Research, 11(3), 411-434.
11. Ikeda, M., Seta, K., Mizoguchi, R. (1997). Task Ontology Makes It Easier To Use Authoring Tools. In Proceedings of IJCAI 1997 Conference.
12. Inaba, A., Supnithi, T., Ikeda, M., Mizoguchi, R., & Toyoda, J., (2000). How Can We Form Effective Collaborative LearnningGroups - Theoretical justification of Opportunistic Group Formation with ontological engineering. In Proceedings of ITS 2000 Conference, 282-291.
13. Mizoguchi, R., & Bourdeau, J. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems, International Journal of AIED, 11(2), 107-121.
14. Murray, T. (2003a). Principles for pedagogy-oriented Knowledge-based Tutor Authoring Systems. In Murray, Ainsworth, & Blessing (eds.), Authoring Tools for Advanced Technology Learning Environments, Kluwer Academic Publishers, 439–466.
15. Murray, T. (2003b). An Overview of ITS authoring tools. In Murray, Ainsworth, & Blessing (eds.), Authoring Tools for Advanced Technology Learning Environments, Kluwer Publishers, 491–544.
16. Nkambou, R., Gauthier, G., Frasson, C. (1996). CREAM-Tools: An Authoring Environment for Curriculum and Course Building in an ITS. Computer Aided Learning and Instruction in Science and Engineering, 186-194.
17. Redfield, C. L. (1997). An ITS Authoring Tools: Experimental Advanced Instructional Design Advisor, AAAI Fall Symposium, 72-82.
18. Ritter, S., Blessing, S., & Wheeler, L. (2003). Authoring tools for Component-based Learning Environments. In Murray, Ainsworth, & Blessing (eds.), Authoring Tools for Advanced Technology Learning Environments, Kluwer Academic Publishers, 467-489.
19. Soldatova, L., & Mizoguchi, R. (2003). Ontology of tests. Proc. Computers and Advanced Technology in Education, In Proceedings of CATE 2003 Conference, 175-180.
20. Supnithi, T., Inaba, A., Ikeda, M., Toyoda, J., & Mizoguchi, R. (1999) Learning Goal Ontology Supported by LearningTheories for Opportunistic Group Formation, Proc. of AIED'99, Le Mans France, 67-74.
21. Vassileva, J. (1995). Dynamic Courseware Generation: at the Cross Point of CAL, ITS and Authoring. In Proceedings of ICCE 1995 Conference, 290-297.