Process-aware Authoring of Web-based Educational Systems

Lora AROYO[1] and Riichiro MIZOGUCHI[2]

[1] Computer Science Department, 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
miz@ei.sanken.osaka-u.ac.jp

Abstract. In this paper we discuss how the concept of ontology can be beneficial for the authoring support of Web-based educational systems (WBES). We take a semantic perspective on the knowledge representation within such systems and explore the interoperability between the various ontological structures for domain and instructional modeling and the modeling of the entire authoring process. This work is based on previous research on modeling of application-oriented aspects within two existing web-based systems SmartTrainer [10] and AIMS [2]. Authoring Task Ontology (ATO) is envisioned as main driving mechanism in providing knowledge to the authoring tools and in enabling them to guide, support, navigate and check the consistency and completeness of the authoring activities at all levels of complexity. It provides operational semantics to the authoring process, by means of specification of authoring activities, sub-activities, goals and stages. In the construction of ATO we consider specification of top-level authoring activities, low-level authoring functions, decomposition of the authoring process in sequence of phases and decomposition of the authoring process in main activity modules (related to the educational system components). Further we present design principle of the ontology-based framework for authoring support of WBES.

1. Introduction

The high and dynamic user demands in many aspects of software production are influencing also the research in the field of knowledge-based educational software and instructional environments. The ultimate problems are related to keeping up with the constant requirements for content flexibility and adaptability, for learning objects and structures reusability and sharing [11]. Most of the limitations of the current knowledge-based educational systems lie in the scope of constructing the sophisticated knowledge base and well-founded application modeling. Many of the systems also implement only a single tutoring and instructional strategy and do not consider much of adaptation within the educational mechanisms [1]. The user modeling is also tutoring restrictive and does not always support various types of procedural and informational skills of the students. The flexibility with respect to upgrading or applying changes to the content, functional components and the knowledge parts of the systems are quite cumbersome and difficult [16]. From the analysis of the current state of the art of AIED research [15] it appears that there is a deep conceptual gap between authoring systems and the authors. The authoring tools are not intelligent and not very user-friendly. Some special-purpose tutoring systems provide quite extensive authoring guidance, but the disadvantage here is that changing or updating such systems is not very easy, and all the knowledge and content can hardly be reused for other educational purposes [16]. The high demands and complexity of the systems reflect also on the authoring environments themselves no matter whether they are general purpose or application oriented. [9,17]. Many of the authoring activities, which the author should perform, are also not semantically clear and do not provide ways to check and assess the authoring result.

All this leads to the requirement that the authoring tool has to offer tunable complexity and autonomous performance of authoring tasks. Structured guidance and feedback to the authors in the complex scale of the design process is also needed.

Currently, a considerable amount of the research on knowledge-based and intelligent systems moves towards concepts and ontologies [12,15] and focuses on knowledge sharing and reusability [10,13]. Research focused on ontologies offers tools and technologies in this respect and helps web-based educational systems move towards semantics-aware environments. In general ontology is used to define the basic terms and relations in the domain. Next to this it also provides axioms as rules and constraints for manipulating and managing of the terms and their relations within this common domain vocabulary. Ontologies allow the definition of an infrastructure for integrating intelligent systems at the knowledge level, independent of particular implementations, thus enabling knowledge sharing [7]. Together with various reasoning modules and common knowledge representation techniques, ontologies can be used as the basis for development of libraries of shareable and reusable knowledge modules (which take the form of software components) [11].

In this context, we consider systematization of the authoring process activities in authoring task ontology (ATO) to support better process analysis and more efficient support for knowledge representation within authoring environments. It provides both methodology and vocabulary [14] for better description of the authoring process and for more effective reasoning. In order to exemplify the ideas of this paper we refer to two Web-Based Educational Systems designed and built independently, but still sharing some common architectural features. Given the complexity and similarity of their authoring tasks we came to the conclusion to propose a common authoring support framework based on ATO, which will provide operational semantics to the authoring process, by means of specification of authoring activities, sub-activities, goals and stages. ATO specifies both top-level authoring activities and low-level authoring functions, decomposition of the authoring process in sequence of phases and decomposition of the authoring process in main activity modules (related to the educational system components) [3,4]. There is no doubt that the ontological structure of ATO is quite suitable for the specification of the authoring activities work on data elements, which are well describable as data and although authoring is a process the functions are working over parameters, which belong already to an ontological structure. This will allow for better interoperability and reusability in the future. Next to this, we will also consider using of Instructional Theories Ontology in order to provide more adequate instructional support and guidance [5].

2. Web-Based Educational Systems

2.1 Web-Based Educational Systems Reference Architecture

As we saw, the process of designing and constructing WBES is mainly handcrafting and very application specific. However, many of those knowledge-based systems share common architecture features, which allows to specify a somewhat reference architecture for concept-based intelligent educational systems. In this way we can allow for more structured and common approach in the authoring process, as well as supporting the automation of the authoring activities. In the analysis of various educational systems [16], such as AIMS [2], InterBook [8], AHA! [6] observe that providing user-oriented instruction depends on three factors:

- The instructional system must be based on a *Domain Model*, describing the structure of the information content within the system (based on concepts (domain terms) and the links between them).
- The instructional system must maintain a *User Model* that represents a user's preferences, knowledge, goals, activity history, current state and other relevant aspects from instructional point of view (often as an overlay model of the domain model).
- The system must be able to provide an adequate *instructional guidance*, coaching and teaching, which is adapted to the user preferences, current status and knowledge level (presented in the UM) within the stated domain model. In order to do so the author must provide an Instructional Model consisting of instructional rules and ontology. The ontology provides vocabulary and concepts, axioms necessary to describe the parts of the instructional model. The rules the process of generating user-oriented view (the adaptation on the UM and the instructional strategy).

The division into a domain model (DM), user model (UM) and instructional model (IM) provides a clear separation of concerns when developing an Web-Based Educational Systems. Our reference architecture advocates the separation of these three components in any Web-Based Educational Systems, aiming at providing user-oriented instruction. With this separation the design of each part of the instructional system becomes clearer and allows the system more flexibility in the updating process. By introducing a reference model we provide a reference to compare different Web-Based Educational Systems by trying to translate the instructional system in the terms of this reference model.

2.2 Authoring Architecture for Web-Based Educational Systems

In relation with the reference architecture we aim at defining an authoring architecture for Web-Based Educational Systems. In Figure 1 we present an outline of this intention by exemplifying it with SmartTrainer as an example of a CBT system and with AIMS as an example of intelligent information management system. The educational process is initiated by the author who supported by an authoring assistant transforms the raw learning material into well structured content and instructional rules to be applied on it. In this manner the author creates (or alters already existing) domain model, constructs the desired learner's model and generates alternative of learning paths within the general instructional model in order to achieve specific instructional goals. The student on the other hand uses the prepared by the author learning material and instantiates his User Model in order to proceed further with the learning process.

In the case of AIMS [2] the Domain Model construction involves semantic representation of the domain terminology in a network of concepts and specification of link types between the concepts. The UM construction here involves construction of an overlay of the Domain Model in a network of concepts with a set of attributes for each concept in order to indicate what is the current status. Finally, the Instructional Model construction involves specification of course topics hierarchy and course tasks sequence within the defined instructional rules. This leads to a conceptual specification of learning items realized with annotated documents linked to each of the course topics and tasks.

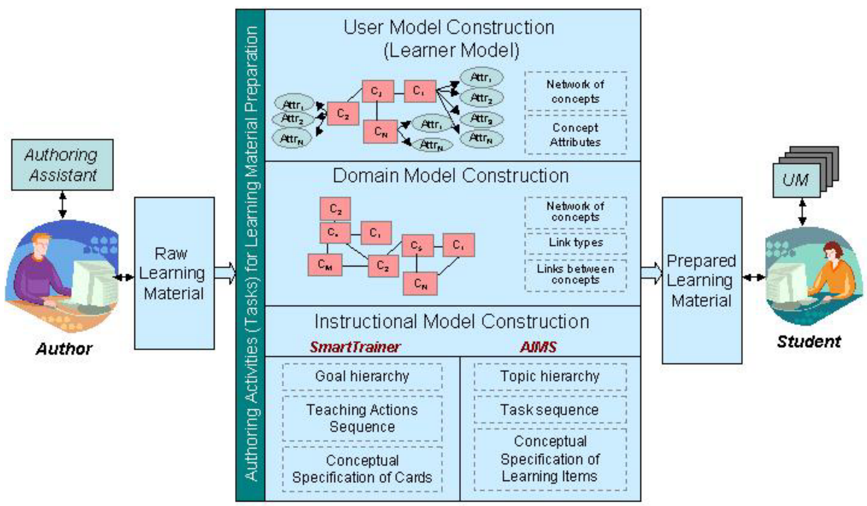

Fig. 1. An Outline of the Authoring Process based on the Web-Based Educational Systems Reference Model

In the case of SmartTrainer the authoring activities by the Instructional Model construction involve learning goal hierarchy construction, teaching actions sequence definition and conceptual specification of cards. These activities are related to the B-model definition within the training task ontology in SmartTrainer [10].

In this process an important part plays the Instructional Model, empowered by an Instruction Engine, by the navigation of the authoring process and providing semantics to it. The Instructional Engine executes instructional rules over the domain model ontology. It consists of an Operation and an Assistant layer [4], the interaction of which results in authors support. The Operation layer handles the operations related to data in each of the reference model components (DM, UM and IM) thus providing means for modeling data and creating alternative goal-oriented structures for teaching.

It consists of engines to operate over the domain model, user model and instructional model. In this layer we construct functions related to information manipulation, consistency of operations and completeness of the process. They deal with tasks such as: handling notions of semantic equivalence and conflict, handling conflict resolution rules, handling equivalence comparison rules, enhancing the resulting ontology and defining additional constraints if necessary. In relation to the reusability support, we pay special attention to the issues associated with merging models, such as extracting portions of the domain, user and instructional models to be merged with another, identifying which frames are to be extracted from the source structure, determining if the extracted information has semantic overlaps or conflicts with the target structure, assisting in merging models, recording the sources of inserted sub-structures for later reference and update, selecting patterns, templates to present them as predefined objects for other authors. Important issues here are also related to the following types of semantic overlaps and conflicts for semantically equivalent concepts but with different names, for semantically different concepts but with the same name, for semantically equivalent concepts with the same name but different definitions and for semantically equivalent concepts linked to different (sometimes conflicting) concepts, etc.

While the Operation layer actually implements the authoring operations, the Assisting layer is directly responsible for interpreting the results from the Operation layer and helping the author by giving hints for how to create a course structure, or how to link a document to the domain model, or how to link a teaching item to the domain model ontology, etc. With the help of the semantic definition of the whole authoring process the assisting layer has a better understanding of the process and possible conflict situations.

3. Ontology-based Authoring Support Framework

According to [15], the authoring process knowledge can be split in two parts - static and dynamic knowledge organization. The static one corresponds to the curriculum organization with Instructional Design (ID) and the dynamic one reflects the tutoring strategy organization for adaptation to the learner. Such a separation is important and is in compliance with the main goal of modularizing the different parts of instructional systems and the process of their authoring. This way we can achieve the desired separation of concerns between data, application and final presentation of the instruction. In this paper we discuss the construction and the benefit of authoring framework to enable the support of the entire authoring process. It includes both knowledge on instructional design and on tutoring strategies in an ontological synthesis. Figure 2 shows architecture of Authoring Support Framework for Web-Based Educational Systems.

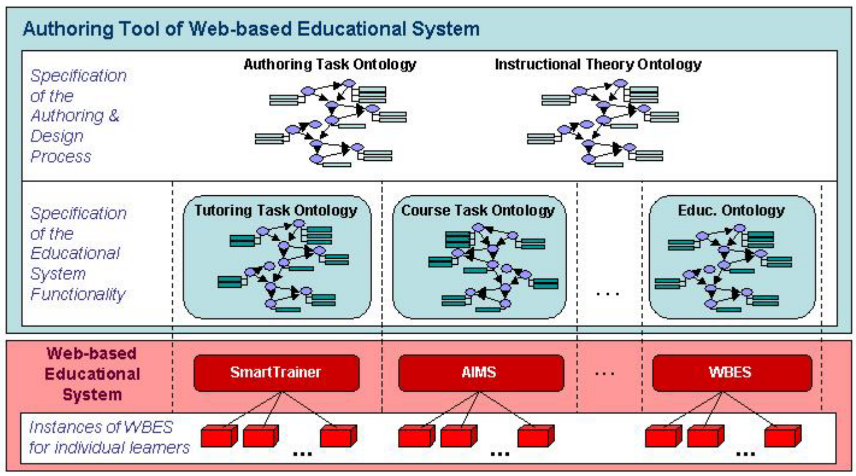

Fig. 2. Ontology Layers

In order to achieve better understanding and clarity of the proposed approach we use two instructional systems *SmartTrainer*, developed at Osaka University, Japan, and *AIMS*, developed at University of Twente, The Netherlands, to exemplify this framework. The bottom part of the Figure 2 depicts the educational system with its type and various instances for each user (depending on the teaching strategy and goal and the UM). In other words, after identifying the educational system by its teaching strategy (type) we look at the instance, which is created in the moment each individual learners uses the system and supplies information for his user model. Further, in the upper part of the same figure, we build the authoring tool for the corresponding educational systems. The knowledge, which we supply to the authoring tool is split in two layers (in correspondence with [15] e.g. dynamic part, where a specification of the educational system functionality is given and a static part, where a specification of the authoring and design process is given in compliance with instructional design and instructional theories. The former helps the authoring tool to understand how the instructional system works (e.g. knowledge about the tutoring strategy and the learner's adaptation). The latter allows the authoring tool to reason on a higher level of curriculum and instructional theories and design. The ontologies in the two layers of the authoring support framework are constructed in correspondence with the three ontology levels introduced by [15]. This means that *level 1* provides set of terms as a well-structured shared vocabulary in order to specify the instructional functionality; *level 2* describes the relationships between the terms and define semantic constraints in the form of axioms. This level builds the major part of the intelligence within the authoring tool. And finally in *level 3* we describe the meta-model of the authoring process interrelated with a systematization of instructional theories. This way, the ontology concept appears to be suitable solution for knowledge systematization within the authoring support tools.

4. Task Ontology for Web-Based Educational Systems Authoring

Web-Based Educational Systems authoring task ontology presented in this paper is based on the notion of 'task ontology' defined by [14]. It provides a system of vocabulary for describing activity oriented structure of all existing tasks within the WBES authoring in a domain independent way. With the help of the authoring task ontology we are able to formalize the tasks involved in the WBES authoring process and specify and modularize all the authoring activities within their specific context. This way a link can be made to the various roles (viewpoints) of the author. Reusability of components and content will be enabled this way as well. Finally we are able to translate the system functions into semantic knowledge in order to standardize the process of WBES authoring, with respect to teaching material, learner model and educational effect.

The authoring process stands for a collection of various authoring activities over domain objects, where the process of their sequencing and combination is guided by axioms and constraints provided by instructional design and each domain object carries a specific role within the corresponding authoring activity. The authoring activities are independent of the system's domain, the educational strategy and the educational goal. The authoring task ontology describes the relations among the authoring tasks and the roles of domain objects, which they play in a particular task. Each authoring task is defined by: (1) sequence of activities with their activity type, constraints and input/output resources, with their resource type and constraints; (2) goal; (3) requirements; (4) constraints, which can be flexible or hard constraints. Following this definition we define for the task ontology of Web-Based Educational Systems authoring the following set of (1) generic nouns reflecting the roles of the objects in the authoring process, (2) generic verbs representing authoring activities over the objects, (3) generic adjectives representing the modifications of the objects and (4) other authoring task specific concepts.

Nouns: CONCEPT, LEARNING ACTIVITY, STRUCTURE, RESOURCE, COURSE, LESSON, DOMAIN, AUTHOR, STUDENT, TEXT, RELATIONSHIP, GOAL, CONSTRAINTS, etc.

Verbs: which are applied in a combination or sequence of activities with specific objects or concepts and their modifications. They are also defined as a set of procedures representing its operational meaning. MODIFY, EDIT, ASSIGN RETURN, UPDATE, SELECT, CHECK, etc.

Adjectives: applied for modification and identification of objects' attributes. SHARED, FINISHED, REQUIRED, IDLE, IN-USE, UPDATED, etc.

Other concepts: CONCEPT PREREQUISITE, LESSON CONSTRAINT, CONSTRAINT SATISFACTION, STATE, ATTRIBUTE, PREDICATE, KNOWLEDGE, etc.

4.1 Definition of primitive tasks/functions

The primitive functions are defined on objects (e.g. concepts, documents, course topics and tasks, etc.) within a specific structure such as DM, UM, IM (e.g. course structure, goal hierarchy, teaching action sequence, etc). They express a simple functional formalism, where the object changes the structure, or the structure is manipulated. Examples of atomic authoring functions include:

CREATE (Structure), CREATE (Object, Structure), ADD (Object, Structure)

DELETE (Structure), DELETE (Object, Structure), VIEW (Objects), EDIT (Object, Structure), LIST (Objects, Structure), UPDATE (Object, Structure), UPDATE (Structure), where object ∈ {{Domain_Concepts} ∪ {Course_Topics} ∪ {Course_Tasks}∪{Library_Docs}} and Structure ∈ {Domain_Model, Course_Model, Library_Base}. Note that such a definition is independent of the structure – the only prerequisite for it is to be concept-based.

4.2 Definition of higher level authoring tasks

At this stage we aim at defining hierarchy of higher-level functions to represent conceptual categories of relationship (interdependence) between primitive functions.

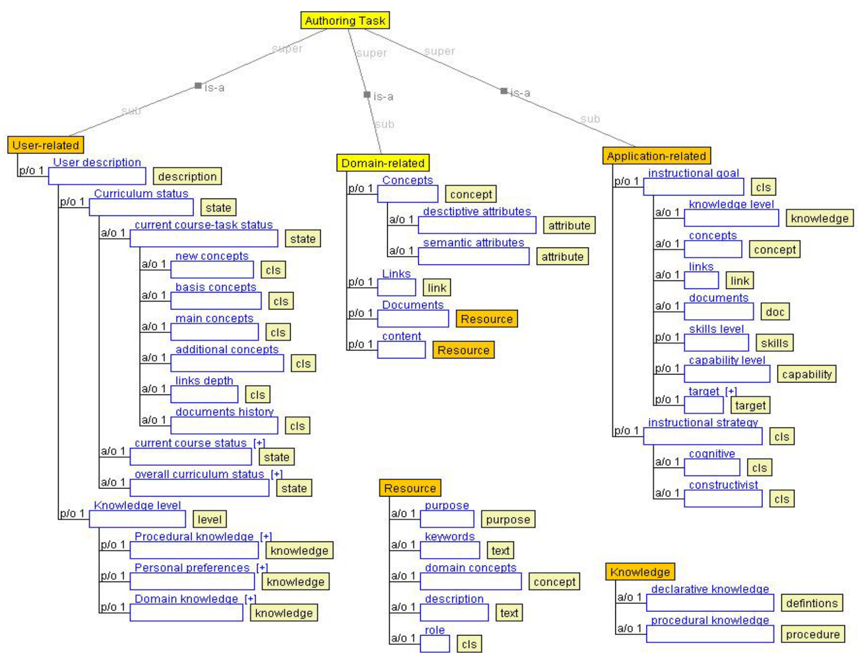

Fig. 3. Fragment of the ATO ontology

These present certain aggregation criteria (including causal and other relations among components) that are used for grouping primitive functional concepts into higher-level authoring functions (classes). This way we can construct/identify functional groups of authoring tasks. The higher-level functions represent a role of one base function for another base-function. They are concerned not with the actual change in the objects, but with their actual function in the process of authoring Web-Based Educational Systems. We define those functions with conditions for their primitive parameters in order to achieve specific authoring goals. This will be based on extracting the functional structure for Web-Based Educational Systems authoring from existing authoring models (as domain) and their connection to educational information. Examples of links in this layer include: 'is-a-prerequisite-for','is-assigned-to','is-achieved-by','follows-from','is-preceded-by','requires','is-followed-by','if-<goal>-then-<action>'

Examples of some common higher-level authoring tasks are:

- DeleteLinks (C_1, DM) - delete all direct links of concept C_1 within the domain DM
- DeletePath (C_1, C_2, DM) - delete all concepts on a direct path between C_1 and C_2 in the domain model DM. These actions can be implemented with a repetitive call to atomic operations, in this case Del(C_i,DM) and Del(L_j,DM).
- DeleteAll (T_o, CS), DeleteAll (T_a, CS) – applied to course tasks T_a or topics T_o in a specific course structure CS. These actions are implemented with a repetitive call to atomic operations, such as Del(T_o,CS_i) – delete all the topics T_o in course structure CS, and List([T_{o1},…, T_{oN}],CS_i) – list all the topics [T_{o1},…, T_{oN}] in the .

Other examples of higher-level authoring tasks are: Link(Doc_j,T_{ik},CS), Link(Doc_j,C_{ik},DM), Copy(CS_{ij},CS_{km}), Compare(Obj_1,Obj_2,Structure), Exist(Obj_i,Structure).

We identify three main groups of authoring, related to the domain model, user model and instructional model construction [Fig. 1]). Within these we define a hierarchical organisation of concepts linked by the ontological link types 'is-a', 'part-of' (p/o) and 'attribute_of' (a/o) [15]. A fragment of the ATO is in Figure 3.

5. Conclusions

The main purpose of this paper is to illustrate the benefits of ontology-based authoring framework for Web-Based Educational Systems in order to achieve author-friendly primitives in terms of which the author can easily describe the goals and methods of their teaching within Web-Based Educational Systems. We discussed the basic issues related to the concept of authoring task ontology, which attempts to systematize the authoring process knowledge and enable the authoring tools to reason more efficiently and provide more user-friendly interaction. Further we give the design principle of a common ontology-based authoring framework for Web-Based Educational Systems, where ATO is the main trigger for the authoring knowledge. Examples of ATO tasks are given within the scope of AIMS and SmartTrainer educational systems. Future work involves complete definition of ATO and its further implementation within the common authoring framework.

Acknowledgements

This research has been performed in a research collaboration at MizLab, Osaka University, Japan. The authors are grateful to Mitsuru Ikeda for his valuable comments and to Yusuke Hayashi and Kouij Kozaki for their support.

References

1. Anderson, J.R., Boyle, C.F., Corbett, A.T., Lewis, M.W. (1990). Cognitive modelling and intelligent tutoring. Artificial Intelligence, 42 (1), 7-49.
2. Aroyo L., Dicheva D. (2001). AIMS: Learning and Teaching Support for WWW-based Education. *IJCEELL*, 11(1/2), 152-164.
3. Aroyo, L., Dicheva, D. (2002). Authoring Framework for Concept-based Web Information Systems, In Proc. of ICCE'02 Workshop, Workshop on Concepts and Ontologies in Web-based Educational Systems, Auckland, New Zealand.
4. Aroyo, L., Dicheva, D., Cristea, A. (2002). Ontological Support for Web Courseware Authoring. In Proc. of ITS'02 Conference, Biarritz, France.
5. Bourdeau, J., Mizoguchi, R. (2000). Collaborative Ontological Engineering of Instructional Design Knowledge for an ITS Authoring Environment, In Proc.of ITS'00 Conference.
6. De Bra, P. and Calvi, L., AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, pp. 115-139, Taylor Graham Publishers, 1998.
7. Breuker, J., Bredeweg, B. (1999). Ontological Modelling for Designing Educational Systems, Workshop on Ontologies for IES at AIED'99, Le Mans, France.
8. Brusilovsky, P., Schwarz, E., Weber, G. (1996). A Tool for Developing Adaptive Electronic Text-books on WWW, In Proc. of WebNet'96, San Francisco, CA, USA.
9. Brusilovsky, P., Ritter, S., Schwarz, E. (1997). Distributed intelligent tutoring on the Web. *AI-ED*, Amsterdam: IOS Press, 482-489.
10. Chen, W. Hayashi, Y., Kin, L. Ikeda, M. and Mizoguchi, R. (1998) Ontological Issues in an Intelligent Authoring Tool, in Chan T-W., Collins A. & Lin J. (Eds.), In Proc. of ICCE'98 Conference, 1, 41-50.
11. Devedzic, V., Jerinic, L., Radovic, D. (2000). The GET-BITS Model of Intelligent Tutoring Systems. *Journal of Interactive Learning Research*, 11(3), 411-434.
12. Devedzic, V. (2001). The Semantic Web - Implications for Teaching and Learning, In Proc. of ICCE 2001Conference, Seoul, Korea.
13. Ikeda, M., Seta, K., Mizoguchi, R. (1997). Task Ontology Makes It Easier To Use Authoring Tools. In Proc. of IJCAI'97 Conference.
14. Mizoguchi, R., Sinitsa, K., Ikeda, M. (1996). Task ontology design for intelligent educational/training systems, Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs. In Proc. of ITS'96 Conference, Montreal, Canada.
15. Mizoguchi, R., Bourdeau, J. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems, *International Journal of AIED*, 11(2), 107-121.
16. Murray, T. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10, 98-129.
17. 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*, LNCS 1108, 186-194.