OntoAIMS: Ontological Approach to Courseware Authoring

Lora Aroyo [1,3]
[1] Department of Mathematics and Computer Science, Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
l.m.aroyo@tue.nl

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

Christian Tzolov [3]
[3] Department of Computer Science, University of Twente
P.O. BOX 217, 7500 AE Enschede, The Netherlands
tzolov@cs.utwente.nl

Abstract: In this paper we discuss how current ontology concepts can be beneficial for more flexible and semantic rich description of the authoring process and for the provision of authoring support of Intelligent Educational Systems (IES) with respect to the three main authoring modules: domain editing, course composition and resource management. We take a semantic perspective on the knowledge representation within such systems and explore the interoperability between the various ontological structures for domain, instructional and resource modeling and the modeling of the entire authoring process. We build upon our research on Authoring Task Ontology and exemplify it within OntoAIMS system. We present authoring scenarios and show their mapping with authoring task ontology. Further we discuss the OntoAIMS framework for management of electronic learning objects (resources) and their usage in the automatic generation of course templates for the authors. Finally, we describe our architecture, based on the ontological specification of the authoring process.

1. Introduction

The use of Intelligent Educational Systems (IES) in many aspects of education has proven its necessity and its efficiency in reaching better learning results (Anderson 1992, Nakabayashi et al. 1996, Vassileva 1995). However, there is still a major gap between the actual tools and the authoring environments (Brusilovsky, 1995, Chen et. al. 1998). Authoring environments and tools can very much help for the popularity of IES to make them also open, interoperable and reusable. Many of those systems have been built in an application specific way and prevent the authors to reuse and share existing teaching units or educational content. At the same time those systems share many common components and are targeting similar subject domains (Mizoguchi et.al. 2000). This is getting a very serious problem in the light of the exponential growth of courseware and learning material. Obviously, there is a need for authoring processes, which produce reusable (modular) courseware, emphasizes on the structure and the modularization of the authoring process and is based on existing standards (both for content and educational knowledge) in order to support interoperability (Murray 1999). The current Semantic Web research provides methods, technologies and standards (Devedzic 2001, Lai et al 1995), which can be used to facilitate the richer semantics for both the content and the processes within courseware. By applying semantic web technologies we can directly use the tools already developed (on rather sophisticated level) in order to facilitate the authoring process in courseware. Common specification languages, like RDF, DAML/OIL (OWL), allow for modularization and semantic organization of courseware authoring process and offer openness and interoperability of authoring tools. We believe that ontological organization of educational content as well as the functional knowledge can be beneficial for more user-friendly authoring environments. We base these beliefs on the results of organizing functional knowledge in task ontology (Fukuhara et al 1995, Mizoguchi et al 1996). Thus, we provide a system of vocabularies to describe the activity-based structure of authoring tasks in IES. The benefits of this approach are:

- reusability of authoring components and content is enabled this way;
- system functionality is translated into semantic knowledge in order to standardize the process of intelligent educational systems authoring;
- independent definition of the subject domain, teaching resources, course components, learner model and educational effect;
- definition of separate viewpoints of authors, which can collaborate within the entire process.

We aim for a common conceptual approach to model both IES functional components and the authoring process, based on common authoring language (ATO) to express authoring tasks (Aroyo et. al. 2003).

Section 2 briefly recalls the AIMS authoring model, introduced in (Aroyo et al 2001, 2002). Section 3 extends our definition of ATO by describing in detail the authoring tasks and activities involved. In Section 4 we illustrate authoring scenarios in order to exemplify the authoring process. Further, in Section 5 we present design concerns of the current OntoAIMS authoring implementation. Finally, Section 6 presents a short discussion about the benefits and implications of such an ontology-based authoring approach.

2. AIMS authoring model

The AIMS model is composed of the following components: *domain model* (*DM*), *course composition model* (*CCM*) and *resource management model* (*RMM*). *User model* (*UM*) is an overlay model of the *DM* and guides the selection choices in *CCM* and *RMM*. AIMS uses a strict separation and independency between domain expert and course authoring roles, based on separate definitions of domain and course knowledge. Three groups of authoring tasks exist:

- *Domain-related:* constructing (editing and annotating) the *DM* in terms of common domain-specific terms (concepts and links between them) and defining additional AIMS specific terms, such as description, resource reference, classification, synonyms, knowledge propagation, link weight, etc.
- *Course-related:* generating the *CCM* as a course structure or sequences of course tasks. Each course task consists of atomic (primitive) activities with its attributes. Each course task is related to relevant domain concepts and resources via the definition of atomic activities.
- *Resource-related:* defining the collection of educational resources in the *RMM*. Each resource is enriched with the appropriate AIMS metadata to facilitate its further use within the course construction module.

The *User-related* authoring tasks are currently realized as a subset of the *course-related* authoring tasks. They deal with the definition of *UM* attributes and their application in the course alternatives and in building knowledge value on considered domain concepts. All the authoring tasks are further organized in the *authoring task ontology* (*ATO*), exemplified in Figure 3 and presented in Section 3.

2.1 Domain model definition

The domain model (*DM*) in IES is a semantic graph representation of nodes and labels, where the nodes are set of domain concepts *C* and the relations are set of links *L* between the concepts, *DM*<*C, L*>. Domain concepts in *C* are defined as <C_i, A_{ij}>, where C_i is the name of the concept and A_{ij} is the corresponding set of attributes. Attributes can point to *name, synonyms, description, context of use*, or to level of importance in the *DM*, namely, *'category', 'sub-category', 'topic'* and *'sub-topic'*. The domain links *L* are defined as <C_i, C_j, T_l, W_l, D_l>, where the C_i and C_j are domain concepts connected with *L*, T_l gives the relationship type, W_l gives the relationship strength and D_l specifies the direction of the relationship. Corresponding to *DM* definition we define domain-related authoring activities, such as '*Add concept', 'Edit concept',* etc. Aroyo et al (2001,2003) give an extensive list of domain-related authoring tasks.

2.2 Course composition model definition

The course composition model (*CCM*) is defined as <*Course_Topics (Course_Tasks (Educational_Activities))*> concept structure, where *Educational_Activities* (*EA*) are atomic (primitive) activities, and the *Course_Tasks* are their composites. *EA* is defined by <$Name_A$, In_A, Out_A, Pre_A, $Post_A$>, where In_A and Out_A define the input and output, and Pre_A and $Post_A$ define the pre- and post-condition. For the purposes of OntoAIMS we use a finite predefined set of *EAs*, such as '*read*', '*write*' and *'exercise'*. Course task (*CT*) is defined by <$Name_T$, EA_1, …, EA_N, I_T, Out_T, Pre_T, $Post_T$>, where the input and output, together with pre- and post-conditions for this course task are specified. For example, Input = <*List of Domain Concepts*>, Output = <*updated values of domain concepts*>, etc. *EA*s are providing various alternative sequences for each course task. The conditions for choosing a sequence are defined within the user model of the student. Examples of course tasks are, *'reflect', 'learn', 'design', 'practice'*, etc. (more in Section 3).

2.3 Resource management model definition

Resource management model (*RMM*) is defined by <$Resource_i$, *Operation*, c_1, …, c_N, A_{ij}>. In the context of the *RMM* definition we define resource-related authoring tasks, explained (specified) by a corresponding set of primitive authoring activities, and further abstracted within the *ATO* (Section 3).

Until here we presented the definition of the AIMS authoring model with its three main modules and their components in compliance with the general reference model of IES authoring (Aroyo et al 2003). Further, we proceed to the formalization of the authoring process within these three main modules. On a system configuration level we define set of authoring primitives (atomic activities), which are automatically initialized within the context of either the domain, course or resources module. On composition level, these initialized primitives are translated into authoring activities, which are further used to compose higher-level authoring tasks. The second step in the process of achieving general reference model of IES authoring, is to find the appropriate form in order to be able to deal systematically with this authoring knowledge. We choose ontological form in order to systematize the authoring process knowledge further.

3. Authoring Task Ontology

In this paper we elaborate on previous research (Aroyo et al 2001, 2003, Mizoguchi et al 1996) defining atomic authoring activities and higher-level authoring tasks, by exploring the possibilities of implementing authoring support tools based on the AIMS authoring model. Keeping up with the strict separation of the domain-related, course composition related and resource management related activities, we try to identify common patterns in order to generalize and automate the performance of the authoring system. The system infers possible authoring choices and allows the author decide on the most desired one.

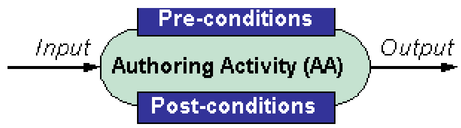

Figure 1. Atomic Authoring Activity definition

With the help of the authoring task ontology we are able to formalize primitive (atomic) authoring activities (Figure 1), which are generic for every authoring process and further compose out of them higher-level and semantic rich authoring tasks, which are the main building block of the overall authoring process. This way we can modularize all the authoring tasks within their specific context of use. The authoring process stands for a collection of various authoring task (Van Marcke 1991) over domain objects, resources and course composition objects, where the process of their sequencing and combination is guided by axioms and constraints provided by instructional design. Each domain object carries a specific role within the corresponding authoring task. The authoring tasks 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 refers to some sequence of activities with their activity type, constraints and input/output resources, with their resource type and constraints. Part of its attributes is defining the goal, the requirements and the constraints (which can be flexible or hard constraints). Each primitive authoring activity specifies some prerequisites (pre-conditions) and some input and produces some post-conditions and some output, which are further used for the selection of the following activity. Chains of atomic activities form/compose the authoring tasks, which are the building blocks of the authoring process. The chains of atomic activities are realized with the following notation:

- Sequence: $AA_1 \wedge AA_2 \ldots \wedge AA_N$
- Choice: $AA_1 \vee AA_2 \ldots \vee AA_N$
- Loop: $AA_1, \ldots, AA_i$, back to AA_1
- If Then: if *Condition* then AA_2 else AA_1

In Figure 2 we illustrate the composition process of authoring activities into authoring tasks.

In order to describe both the atomic authoring activities, as well as their translation in authoring tasks, we define a set of generic terms, which reflect three perspectives on each *authoring object*. These are:

- the *role* of the authoring object within the authoring process. This is described by the usage of generic *nouns*, such as *'concept'*, *'activity'*, *'task'*, *'constraint'*, *'knowledge'*, *attribute'*, *'state'*, *'structure'*, *'resource'*, *'course'*, *'lesson'*, *'domain'*, *'author'*, *'student'*, *'text'*, *'relationship'*, *'goal'*, *'constraints'*, etc.
- the *function* over the object. This is expressed with generic *verbs*, such as: *'edit'*, *'update'*, *'check'*, *'select'*, *'delete'*, *'add'*, *'view'*, *'list'*, *'assign value'*, *'repeat'*, *'focus'*, *'study'*, *'write'*, *'read'*, *'exercise'*, etc. We use these generic verbs to express atomic authoring activities, as well as atomic learning activities. They can be

applied in a combination or sequence of activities over specific objects or concepts in order to express higher-level authoring and learning tasks.

- the *state* of the object. This is expressed with generic *adjectives*, such as *'shared'*, *'finished'*, *'required'*, *'idle'*, *'in-use'*, *'updated'*, etc. The state perspective specifies the modification and identification of objects' attributes.

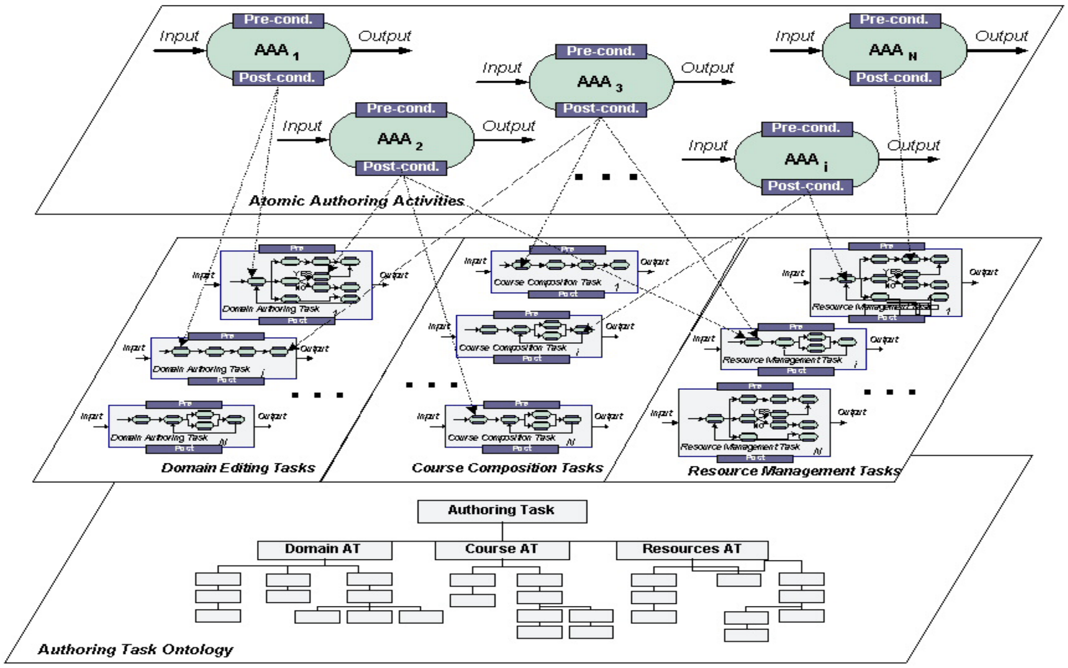

Figure 2. Authoring layers for IES defining ATO

By defining this generic set of terms and specifying the three perspectives we are able to formalize the authoring tasks, as a set (combination or sequence) of atomic activities (expressed with generic verbs) over certain input (expressed with generic nouns) and specified pre-condition state (expressed with generic adjectives). In the same way, we are able to formalize also educational tasks used as building blocks for the instructional model. For instance in a pseudo notation the *CT* '*reflect*' can be expressed with the following:

task_name:reflect {
activity$_1$: *read* $(In_{A1}, Out_{A1}, Pre_{A1}, Post_{A1})$;
activity$_2$: *write* $(In_{A1}, Out_{A2}, Pre_{A2}, Post_{A2})$;
activity$_3$: *exercise* $(In_{A3}, Out_{A3}, Pre_{A3}, Post_{A3})$;
operation: $((activity_1 \wedge activity_2) \vee activity_3)$;
input: $resource_i$, $resource_j$, $resource_n$ *AND* $\{c_{ik}, c_{jm}, c_{nr}\}$;
output: $report_i$ *AND* $\{c_{in}\}$
pre: *resource_state*($resource_i$,*read*) *AND* *knowledge_value*(c_{ij}) = $\{a_{ij}\}$;
postcond: *resource_state*($resource_j$,*written*) *AND* *resource_state*($resource_n$,*exercised*) *AND* *knowledge_value*(c_{ij}) = $\{a_{ij}\}$ *AND* *knowledge_value*(c_{in}) = $\{a_{in}\}+0.70$}

Finally, we are able to achieve modularization of the authoring process and to provide reusable, sharable and interoperable components for IES engineering. The generic set of terms provides vocabulary of concepts in terms of which one can describe the components of IES and the authoring process of IES. And in order to make it usable, we organize it in a task ontology. Following the general definition of ontology, as consisting of two parts – taxonomy and axioms, we are defining the taxonomy of generic terms and define axioms in order to express the design concepts and the constraints within the authoring process. Figure 2 presents the composition process of atomic authoring activities in authoring tasks, by using the vocabulary and the rules defined within the *ATO*. At the high level of abstraction within the task ontology, the domain, course and resource modeling tasks can be viewed as a design tasks, where the typical one is course design. A design task consists of requirement analysis, building an initial structure, evaluate it, identify the causes of problem (where to modify), how to modify, evaluate, etc. The evaluation task requires criteria dependent on instructional theories. The primitive authoring activities explain and implement all the high level authoring tasks in the model. The authoring tasks on higher level also share role concepts such as parts to select and combine, criterion and constraint. The difference at this level is instantiation of the role concepts, e.g. parts in domain modeling are domain concepts, parts in course modeling are learning item and parts in the resource modeling are the resource items. This specification is illustrated Figure 4 and the architectural design presented in Figure 3. There the role concepts, constraints and the criteria are correspondingly defined in domain meta-ontology, course meta-ontology and resource management ontology. These three ontologies define the concepts and give formal specification for the system to understand and map the actual subject domain model (ontology), course structure and resources.

In the following section we will present use scenarios for the OntoAIMS authoring environment. They exemplify the application of ATO within the environment and the authors' guidance within the process.

4. OntoAIMS Scenarios

The AIMS authoring model allows authoring within the three main IES modules – Domain Editing, Course Composition and Resources Management. For each of those system modules a separate working scenario will be presented, performed either in parallel or independent of each other. These scenarios will show how the authoring knowledge in the form of authoring task ontology (e.g. the subset for course composition tasks) can be used to support the construction of course structure within a selected application domain and with a select set of resources.

4.1 Domain editing

The Domain Editing scenario describes the work of the author within the Domain modeling module of OntoAIMS. It outlines all the tasks (domain specific or OntoAIMS domain specific), which the author performs, while attempting to create a suitable domain model in terms of OntoAIMS concepts from an externally created subject domain ontology. This domain ontology defines the common vocabulary of terms to describe the basic knowledge about the selected subject domain. Those terms will be later used as a basis for the composition of the course structure and for the selection of the parts of the domain to be studied by the students. There are two main options for the author to initialize his work within the Domain Editing Module: (1) the author is editing (annotating) existing domain model, or (2) the author is constructing new domain concept model. We consider the second case as not so relevant, as the main intention is to provide reusability of the content, thus to show how the author is adjusting existing knowledge structure to the needs of his course. Important here is the separation between domain specific tasks (e.g. add concept, delete link, etc.), which can be performed within an external ontology editor (or adopted functionality within OntoAIMS, and the OntoAIMS domain specific authoring tasks (e.g. add synonym, annotate concept, set knowledge value, etc.), which are necessary in order the external domain knowledge to fit the needs for course building within OntoAIMS. This distinction is made also on architectural level, by introducing meta-domain ontology (knowledge about the domain authoring concepts in OntoAIMS) and concrete domain ontology (knowledge about the concrete domain in which the course will be given).

4.2 Course composition

The course composition scenario describes the work of the author within the AIMS Course Composition module. Based on the instructional knowledge within OntoAIMS and with initial input from the author about the goals and general description of the targeted course, OntoAIMS Course Editor generates automatically a course structure template, which is further manually filled in by the author with a real content. The final aim here is to define the authoring activities, which the author performs, while attempting to use the template to create a new course structure or to edit an existing one. This involves selecting domain concepts, building a course structure out of them and linking resources to the concepts. A course can cover concepts from different domain models. The author can start working on the course structure only if there is at least one domain concept model. The system-generated template will allow the author to organize all the metadata of his course. Examples of these are: name, description, target group, time duration (number of lectures, seminars, begin and end date), learning goals and assessment strategy of the course and prerequisites for the course (in terms of domain ontology concepts knowledge; and in terms of other courses taken). In order to create the course structure the author should further (1) select concepts from the domain model and assignment them to course topics; (2) select a specific sequence of course topics realizing the learning goals; (3) assign course tasks for each topic (each task will cover more than learning activity); and (4) link educational resources to each course topic. Some concepts can already be linked to resources in the domain. These links (or a subset of those links) may be used in the course, but it's not an obligation. The author should be free to add course specific links and exclude links to resources defined in the domain. This way the author performs three main tasks: (1) structuring of resources and concepts, (2) editing of resources and concepts, and (3) modeling the process of using/executing a course structure. On a general level the author is guided by the authoring system to organize and annotate his educational materials and his intentions to create a specific course. The actual instructional guidance in selecting appropriate pedagogical strategy and the systems monitoring to check whether the current authoring activities realize it, is implemented within the predefined atomic authoring activities and the higher-level authoring tasks.

4.3 Resource definition

The Resource management scenario describes the work of the author within the AIMS resource management module. It will specify and outline all the important activities and task, which the author performs, while attempting to insert new or edit existing resources in the resource repository. This repository is used within various courses and various domains. It can be seen as a general pool of resources for our author, which he can use for all the courses and domain he is working in. The author can include a new document or edit an existing one at any point of time. The scenario presents all the activities related to modeling and indexing information resources, where an information resource could be any possible document or piece of information. The result of the authors work in this module is an indexed list of resources. The authoring steps are as following: (1) build/create a list/pool of new resources. Each resource is described by URI (URL or URN), name, keywords (concepts from the domain ontology), description, context of use (this is still an open question how to be realized); (2) add new resources => give name, URI, description, keywords (domain concepts) of the resources. Next, the author should specify the context of use of the resource (book, article, text fragment, exercise, procedure definition, case, definition, intro text, etc.). All possible educational meaningful formats are considered.

5. OntoAIMS Architectural Description

To describe Onto AIMS architecture (Figure 3) we introduce five roles to interact with the system: the authoring roles of the domain author, resource author and course author, and the user roles of instructor and learner.

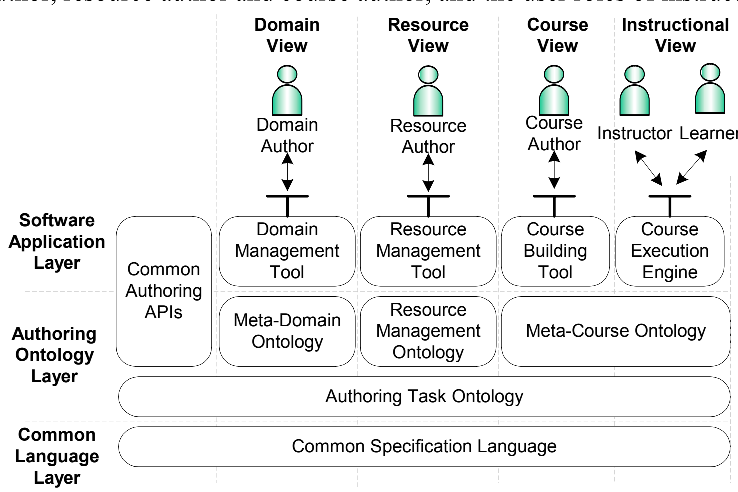

Figure 3. Enterprise view on OntoAIMS

We look at the architecture from two main perspectives:

(1) We distinguish three levels of abstraction defining common specification language, authoring ontologies and the corresponding tools and their users at the software application level.
(2) We define four views or phases of the authoring process, which on the one hand specify the domain, course and resource modeling perspectives and on the other define the user-interaction within the instructional view (both for learners and instructors).

Every user (designer, content expert, author, instructor or student) can play each of the roles in different stages in the process. One and the same user can also perform in all of the roles alone. Each role is defined with the context of the phase and the set of authoring or educational tasks, which are allowed at this point.

5.1 OntoAIMS Layers

The first layer introduces a *Common Specification Language*, shared through out all the phases of the authoring process, which defines the basic concepts and semantics for the specification of OntoAIMS authoring concepts, as well as concepts outside of OntoAIMS, such as ‘resource’, ‘domain’ and ‘knowledge’. This way we ensure the consistency of the system. For the realization of this language we choose DAML/OIL (OWL), as it is web-based, allows for openness and provides large amount of flexibility for reasoning. By using it we can directly use the tools and methodologies designed for general Ontologies and apply them within the context of OntoAIMS.

With the help of the common specification language we define the concepts and the semantics of the authoring process within the upper *Authoring Task Ontology* (*ATO*), where each of the phases (domain, course and resource views) define sub-set of specific authoring concepts (e.g. meta-domain ontology, meta-course ontology and resource management ontology).

The last layer refers to the software applications implementations. A common program interface has been designed (*Common Authoring API*), which is able to process the ATO specification and to capture the common functionality required by the specific authoring tools (e.g. resource, course or domain editors).

5.2 OntoAIMS Authoring Phases

Each authoring phase defines a view for the corresponding role. The roles can be split in two groups (1) roles of the author, such as domain author, course author and resource manager, and (2) roles of the OntoAIMS user, which are represented by instructor and learner. Within each phase we specify all the authoring activities and tasks in terms of ATO concepts and the additional meta-ontologies for the course, domain and resources management. These are further translated into authoring support functionality of the authoring tools provided within each phase. In Figure 4 we present a detailed view on the relations between ATO and the OntoAIMS meta ontologies, as well as the information flow within the process of course composition in OntoAIMS.

5.3 OntoAIMS Computational Model

The OntoAIMS computational model defines the information structures, modules and the flows in the context of course composition process. Figure 4 shows, how the concepts defined on the highest level of ATO are further shared within the meta-level Ontologies for domain, course and resources. Subsequently, specific (concrete) structures are initiating the authoring process in order to finally realize the end goal – course composition.

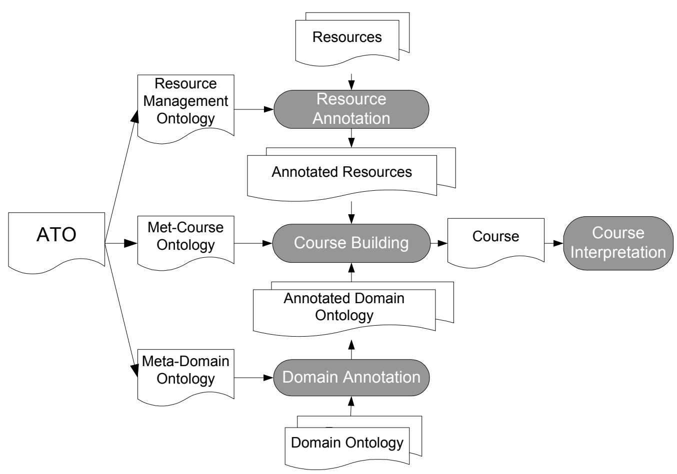

Figure 4. OntoAIMS Computational View

ATO: Authoring task ontology introduces the shared concepts for all the three design tasks. The elements of this ontology are: authoring task, authoring activity, input, output, pre-requisite, post condition. It defines further three met-ontologies, which specify the concrete concepts for each of the OntoAIMS views.

Resources Management Ontology: It defines the specifics of resource creation, editing, maintaining and relating to domain concepts in order to facilitate the process of course composition and its final execution. Elements of this language are: Resource, ResourceURI, name, description, presentation format, instructional format, resource pool, annotation, etc.

Course: Course building process uses the annotated domain and resources and with the help of the met-course language describes the structure and the behavior of the course. The meta-course language elements are: course topic, course task, educational activity, course, sequence and annotation. Here we introduce as well some learning related authoring tasks, such as 'reflect', 'learn', 'design', 'practice', etc.

Meta-Domain Ontology: In order to be able to work with the domain model, as well as with the concrete resources, the author needs to have those resources translated in terms of his own 'expert language'. In the case of OntoAIMS this language is directed by the instructional design theories known by the expert. Following this, Meta-Domain Ontology defines the language for OntoAIMS to enable the understanding of domain concepts, links and other domain primitives. The process of domain model creation (the annotation of an existing ontology in order to create a domain model from it) uses the OntoAIMS terminology, defined by the Meta-Domain Ontology, in order to specify the correspondence between the concepts of the selected domain and instructional concepts needed to build a course within this selected domain. The result is an annotated ontology. Elements of the meta-domain language are: concepts, link, resource reference, description, synonyms, knowledge value, knowledge propagation, other forms, category, sub-category, topic, sub-topic, relationship weight or context of use.

6. Conclusions

In this paper we discussed how current ontology concepts help a more flexible and semantic rich description of the authoring process and provide authoring support for Intelligent Educational Systems (IES). We formalized the IES authoring process in terms of generic tasks and activities within the three main authoring models – domain, course and resources. Here we build upon common IES reference model and research on defining generic tasks. We showed how applying the ontology concept helps us achieve flexibility in reasoning and higher usability of the authoring environments for IES. We have exemplified our ideas in terms of constructing the Authoring Task Ontology (ATO) and implementing it within the OntoAIMS authoring environment. Future work concerns more detailed presentation of ATO and modeling of the OntoAIMS computational view.

7. References

Anderson, J. (1992). Intelligent tutoring and high school mathematics. In Proc. ITS'92.

Aroyo, L, and Mizoguchi, R. (2003). Authoring Support Framework for Intelligent Educational Systems. In Proc. AIED'03 Conference (in print).

Aroyo L., Dicheva D. (2001). AIMS: Learning and Teaching Support for WWW-based Education. IJCEELL, 11(1/2), 152-164.

Aroyo, L., Dicheva, D. & Cristea, A. (2002). Ontological Support for Web Courseware Authoring. In Proc. ITS'02 Conference, 270-279.

Brusilovsky, P. (1995). Intelligent learning environments for programming: The case for integration and adaptation. In Proc. AIED'95, 1-10.

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, 1, 41-50.

Devedzic, V. (2001). The Semantic Web - Implications for Teaching and Learning, In Proc. of ICCE 2001Conference, Seoul, Korea.

Fukuhara, Y., Kimura, F., Kohama, C., and Nakamura, Y. (1995). A Knowledge-based Educational Environment Integrating Conceptual Knowledge and Procedural Knowledge in Telecommunication Service Field. In Proc.ED-MEDIA'95, 229-234.

Kitamura, Y., Sano, T. and Mizoguchi, R. (2000). Functional Understanding based on an Ontology of Functional Concepts. In Proc. PRICAI'00, 723-733.

Lai, M.C, Chen, B. H. and Yuan, S.M. (1995). Toward a new educational environment. In Proc. WWW'95.

Mizoguchi, R., Bourdeau, J. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems, International Journal of AIED, 11(2), 107-121.

Mizoguchi, R., Sinitsa, K., and Ikeda, M. (1996). Task ontology design for intelligent educational/training systems. In Proc. of Architectures and Methods for designing Cost-Effective and Reusable ITSs Workshop, at ITS'96 Conference.

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.

Nakabayashi, K., et al. (1996). An intelligent tutoring system on the WWW supporting ubteractive simulation environments with a multimedia viewer control mechanism. In Proc. WebNet'96, 366-372.

Van Marcke, K. (1991). A Generic Task Model for Instruction. In Proc. of NATO Advanced Research Workshop on Instructional Design Models for Computer Based Learning Environments, Twente, 1991.

Vassileva J. (1995). Dynamic Courseware Generation: at the Cross Point of CAL, ITS and Authoring. In Proc. ICCE'95, 290-297.