Automatic Authoring of Adaptive Educational Hypermedia

Alexandra I. Cristea

Faculty of Computer Science and Mathematics
Eindhoven University of Technology
Postbus 513, 5600 MB Eindhoven, The Netherlands
Tel: +31-40-247 4350
Fax: +31-40-246-3992
a.i.cristea@tue.nl

Craig Stewart

School of Computer Science and Information Technology
University of Nottingham, Jubilee Campus
Wollaton Road, Nottingham, NG8 1BB, UK
Tel: +44 115 846 6505
Fax: +44 115 951 4254
craig.stewart@nottingham.ac.uk

Automatic Authoring of Adaptive Educational Hypermedia

Abstract

Adaptive Hypermedia (AH) can be considered the solution to the problems arising from the "one-size-fits-all" approach to information delivery prevalent throughout the WWW today. Adaptive Educational Hypermedia (AEH) aims to deliver educational content appropriate to each learner, adapted to their preference and educational background. The development of AEH authoring tools has lagged behind that of delivery systems. Recently AEH authoring has come to the fore, with the aim of automating the complex task of AEH authoring, not only within a system but porting material between different AEHs. Advances in *intra*-system automation are described using the LAOS framework, whereby an author is only required to create a small amount of educational material which then automatically propagates throughout the system. Advances in *inter*-system conversions are also described; the aim is to move away from a "create once, use once" authoring paradigm, currently in force with most AEH systems, towards a "create once, use many" paradigm. The goal is to allow authors to use their content in the AEH delivery system of their choice, irrespective of the original authoring environment. As a step along this road we describe the usage of a single authoring environment (MOT) to deliver content in three independently-designed Educational Hypermedia systems (AHA!, WHURLE and SCORM-compliant Blackboard).

This chapter describes therefore advances in automatic authoring and conversion towards a simple and flexible AEH authoring paradigm.

INTRODUCTION TO AEH AUTHORING

Adaptive hypermedia (AH; Brusilovsky, 2001a) started as a spin-off of hypermedia and Intelligent Tutoring Systems (ITS; Murray, 1999). Its goal was to bring the user model capacity of ITSs into hypermedia. However, due to technical limitations, such as bandwidth and time constraints, AH only implemented simple user models. This simplicity also gave AH its power as, suddenly, there were many new application fields and also implementation was considerably easier. Early AH research concentrated on variations of simple techniques for adaptive response to changes in user model. No wonder that most of AH development was research-oriented, applied only to the limited domain of courses the researchers themselves were giving (AHA!, De Bra & Calvi, 1998; Interbook, Brusilovsky et al., 1998; TANGOW, Carro et al., 2001) and with very rare commercial applications (Firefly, developed at MIT Media Lab. and acquired by Microsoft).

Recently there has been a shift in attitudes. The development of the *Semantic Web* (Berners-Lee, 2003), and the ongoing push to develop *Ontologies* (Gruber, 1992) for knowledge domains has extended the importance of AH. Indeed AH now appears to be the tool of choice for collating the static information of these new approaches and bringing then to life.

Moreover, AH is spreading from its traditional application domain, education, to others, especially the commercial realm, which is eager to be able to provide personalization for its customers. Indeed, we often see the phenomenon from other communities re-inventing adaptive hypermedia for their own purposes and applications.

Adaptive Educational Hypermedia (AEH; Brusilovsky, 2001b) is, in principle, superior to regular Educational Hypermedia (EH), as it allows for personalization of the educational experience. Regular EH, such as that delivered by WebCT and Blackboard is not adaptive: exactly the same lesson is delivered to each student. Pedagogical research has shown (Coffield, 2004) that different learners learn in different ways. This is a truth self-evident to most teachers; if a student is having trouble learning a subject, then they will alter the manner in which they are teaching it and try a different approach. Traditional EH systems could be compared to inflexible teachers, who base their lesson mainly on drilling and repetition. Educational systems (real or virtual) that adapt their presentation to the needs of each learner aim to improve the efficiency and effectiveness of the learning process. If each learner has their own Learning Style (Coffield, 2004) and is given a set of resources specific to this particular style then that learner will not only learn 'better', but will be able to more effectively develop the given information into deeper understanding and knowledge. AEH systems seek to address the inflexibility of current EH methods. Systems such as MOT, AHA! and WHURLE all answer the need for an adaptive and flexible approach to teaching. They allow current online educational systems to break away from the "one-size-fits-all" mentality and move towards having an appropriate lesson for each student.

AEH systems aim to improve upon current static EH systems. This is not to say that they are the universal panacea for online education. Education is not undertaken in a vacuum; the social aspect is also vital. It is essential for learners: to be able to build common ground; to ask and answer (negotiate meaning); to argue and debate; to explicate mental models; to share expertise; to collaborate; and to construct novel ideas and understanding. Work on computer-supported cooperative work (CSCW) addresses this side of the educational process, and often AEH systems will fold this research into them (for example WHURLE can be used in such a social manner). Collaborative work can be encouraged by the use of simple online social tools: email, for asynchronous communications; fora, for persistent asynchronous group discussions; and chat rooms, for synchronous group discussions. The addition of Adaptation to this whole structure is another improvement to the student's personal online educational experience.

However, with increasing numbers of students, and the resulting increase in class size of many learning bodies, traditional methods of education (such as the tutorial, and the field trip) often become impractical – in terms of time and cost. Online education can help to fill this need, EH and AEH were developed to do just this.

Given the qualities of AEH systems, it might be reasonable to expect a much wider uptake than actually is happening. A major hindrance of this is that the creation of good quality AEH is not trivial, often involving a greater expenditure of time and money to produce, when compared to standard online educational systems.

Creating content within a single AEH system can be a complex and difficult undertaking.

Many issues must be considered, amongst them:

- *What knowledge domain(s) will the lesson partake of?*
- *Do any previous e-learning materials exist that are both available and re-usable?*
- *What are the objectives of the lesson and how are they to be achieved for a heterogeneous group of learners?*
- *Which traits of a learner are to be modelled and how is this User Model created?*

- *How is the data, concerning these traits, to be gathered, implicitly* (without the learner's knowledge) *or explicitly* (information is requested from the learner)*?*
- *Given that there exists a heterogeneous group of learners how many versions of the same material need to be created?* For example, if a group of learners are to be divided into two sub-groups, one which requires visual materials and the other which requires textual based materials, then it follows that at least two sets of the material are required to teach that lesson.
- *What are the rules for adaptation? Does the author of the lesson have any control over their use or creation?*
- *How are the various versions to be presented to the learner, and does the learner have any control over this?*

Most AEH systems require the author to consider these issues with little or no help. The author is left adrift and often must become an expert in Adaptive Hypertext before creating anything.

It is hardly surprising then, that AEH systems are not used widely outside of their own development circles, as these developers are the only people with the required level of expertise to create content for them! This problem arose whilst AEH was still a new area of research. A natural "one-to-one" paradigm developed, with developers creating the AEH system that was specific to their desires and insights, along with the necessary authoring tools. Cross-platform considerations were not important; transporting data between systems was generally considered irrelevant.

Nowadays, a lot of research effort concentrates on the 'authoring challenge' (Wu et al., 1998; Specht et al., 2001; Murray, 2003; Cristea & Cristea, 2004) in AEH, with the goal of reducing complexity, thereby delivering the greater flexibility of an AEH for the same cost as current online systems. This chapter approaches this challenge from the point of view of *automation*, minimizing, but not restricting, the author's input and reducing overload.

Advances in *inter*-system conversions are also described, the aim being to move away from a "create once, use once" authoring paradigm, as with most AEH systems; towards a "create once, use often" paradigm. The goal is to allow authors to use their content in the AEH system of their choice, irrespective of the original authoring environment. As a step down this road we describe using a single authoring environment (MOT) to deliver content in three independently designed Educational Hypermedia systems (AHA!, WHURLE and SCORM compliant Blackboard).

The remainder of this chapter is organized as follows. First we present LAOS, a generic AH authoring framework that incorporates several layers of semantics to better express the authored AEH. The major part of this chapter focuses on the two major dimensions of AEH authoring automation that we have identified: automation within an AEH authoring environment, and automation outside it, comprising conversion between AEH systems. Finally we draw conclusions.

LAOS LAYERED MODEL

The LAOS model (Layered AHS Authoring-Model and Operators; Cristea & De Mooij, 2003c; Figure 1) addresses the issue of AEH authoring complexity by dividing it into subtasks corresponding to five explicit semantic layers of adaptive hypermedia (authoring), that together act as a framework for designing an AEH.

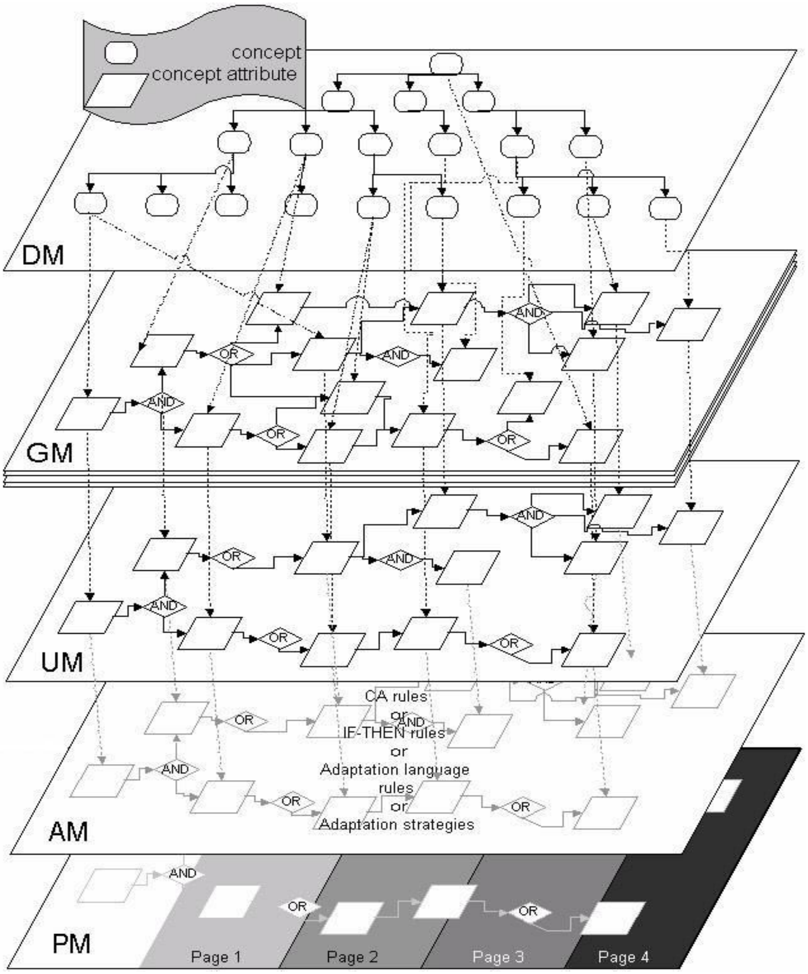

Figure 1. The LAOS Adaptive Hypermedia (Authoring) Framework

These five semantic layers of LAOS are:

- **domain model** (DM), containing the basic *concepts* of the contents, and their representation (such as learning resources)
- **goal and constraints model** (GM), a constrained version of the domain model. The constraints are based on educational goals and motivations.
- **user model** (UM), represents a model of the learner's educational traits.
- **adaptation model** (AM), a more complex layer that determines the dynamics of the AH system. Traditionally, this layer is composed of IF-THEN rules and therefore the LAOS version also translates such rules at the lowest level.

- **presentation model** (PM), is provided to reflect the physical properties and the environment of the presentation; it reflects choices, such as, the appropriate background contrast to support a learner with poor eyesight.

Each of these semantic layers are composed of semantic elements. LAOS allows flexible (re-)composition of the defining semantic elements of the layers, according to each learner's personalization requirements. We are not going to go into details about the semantic elements, except for those directly used in internal automatic transformations or external conversion. At this point, it suffices to remark that the LAOS structure simply serves to make explicit the complex layers of an AEH system.

Such a detailed structure requires a lot of time to populate with AEH instances. As an alternative, we discuss semi-automatic authoring techniques, which populate the whole structure based on a small initial subset that has been authored by a human. Here we analyze two different possible initial subsets:

- *internal semi-automatic authoring*: the theoretical analysis of the semi-automatic generation of one LAOS layer based on (the content and structure of) another one. The practical analysis of this is performed in MOT (My Online Teacher, Cristea & De Mooij, 2003d). In short, we see this research line as another step towards adaptive hypermedia that '*writes itself*'.
- *external semi-automatic authoring*: the theoretical and practical analysis of conversions between AEH authoring systems, such as MOT, into AEH delivery systems, such as AHA! and WHURLE (Moore, 2001) or educational systems, such as Blackboard. We examine the structures resulting from using a single authoring system to convert content for use in each system. In effect

we propose a paradigm shift for AEH authoring, away from "write once, use once" (i.e., every AEH has its own authoring systems) towards a middleware system that allows for delivery of the same material to many different AEHs. We describe the current "state of the art" towards this goal – using MOT as an authoring environment to deliver adaptive content to WHURLE and AHA! (also the connection to Blackboard).

TRANSFORMATIONS WITHIN AN AEH SYSTEM

Adaptive Educational material is obviously more difficult to create than linear educational material, because of the alternative content versions and path descriptions. Therefore, we investigate the possibilities of automatic generation of some of the LAOS layers, using information from other layers. In the following sections, we will sketch some of these transformations, focussing on their semantics. The flexibility of general transformations has been addressed in (Cristea, 2004)

From Domain Model to itself (DM→DM)

The DM contains the learning resources of the AEH, such as the actual course materials, figures, graphs, videos, etc. These resources are grouped under the domain concept they belong to, using the established domain semantics. That is, resources are grouped into *attributes* of given (rhetoric or other) *types*, such as 'text', 'introduction', 'conclusion', 'figure' etc. The DM also contains the *links* between the semantic wrappers of the domain resources, such as links between concepts, grouping them into concept *hierarchies*, or other *relatedness links*. This section discusses the way in which the DM can be automatically (adaptively, adaptably) enriched, by interpreting the semantics of its structure and contents.

New semantic links. The easiest way to enrich the domain model is by automatically finding new domain links between existing domain concepts[1].

For instance, new *relatedness relations* can be generated for relations between concepts that share a common topic. This commonality can be computed at concept attribute level, and therefore can automatically be labelled with a type that corresponds to the type attribute of the connecting attribute. In the following, we illustrate this with the help of an abstract example:

Consider, we have two domain concepts from two possibly different domain concept maps, $c1 \in C1$, $c2 \in C2$ (concept 'NN Introduction' and concept 'The biological neuron' from the concept maps 'Neural Networks I' and 'Neural Networks II' respectively[2]). Now consider now two respective attributes of these concepts $a1 \in c1$, $a2 \in c2$; these attributes can be given as pairs of variable names and their respective values: *a1=<var1,val1>*, *a2=<var2,val2>*. If the attributes are of the same type (*var1= var2=var*; for instance, var='keywords') then a **weighted, typed semantic domain link** can be generated between the two concepts *c1* and *c2*, with the link type (label) given by the type of the attribute, and the weight defined as the number of common features between the two value fields: *weight=number_common_features(val1,val2)*. This link will only make sense if the weight is positive.

[1] These new links can be between the concepts of the *current content* (concept map: e.g., course), between the current content and some *other content created by the same author*, or finally between the current content and some *other content created by a different author*.

[2] Examples taken from LAOS implementation in MOT.

This is one semantically explicit, symbolic way of generating new links between domain concepts. Another way is, for instance, to apply an algorithm that checks the domain map for missing link types, and prompts the author, asking if new ones should be searched for.

New semantic attributes. A different method to enrich the domain model involves link analysis to compare semantically similar concepts (semantically similar can mean similar from a link-point of view, such as concepts sharing the same ancestor-concept, for example; or concepts at the same level of the hierarchy; or concepts related with each other via some special link (of a given type), etc) and to find if some attributes (or even sub-concepts) are missing.

For instance, consider a concept called 'Discrete Neuron Perceptrons' from a Neural Networks course that has an attribute of the type 'Example', whereas the concept 'Continuous Neuron Perceptrons' doesn't, although they are linked via a relatedness relation as described in the previous sub-section.

In this case, the system can signal the author concerning the possible 'missing' content item, corresponding to the semantics (attribute, sub-concept, etc.). It may even look for possible candidates for the 'Example' attribute via other links to this concept. This search space is not limited in scope but can continue '*outside'* the LAOS model, leading to a transition from a closed space to the Open Adaptive Educational Hypermedia space.

From Domain - to Goal and Constraints Model (DM→GM).

The Goal and Constraints Model filters, constrains and restructures the Domain Model, corresponding to a *pedagogic goal*. For instance, a lesson aimed at *beginners* starts by filtering the necessary introductory information from a larger pool defined by one or more appropriate domain maps. Therefore, the *primary content* of the GM are not resources, but copies of (or, rather, to avoid redundancy, pointers to) the resources. The Goal and Constraints Model also contains prerequisite relationships that establish the general recommended order of visiting the course items. Moreover, here the differentiation is made between alternative content (OR relations) and obligatory content (AND relations). Therefore, the GM Model contains mainly structural elements, or links. The GM can also contain resources, if these are of a pedagogical nature only (such as a text explaining why it is better for beginners to study resources, grouped as attributes, with the type 'Introduction').

Automatic (adaptive, adaptable) Goal and Constraints Model enrichment or creation based on the Domain Model can be achieved based on semantic presentation constraints or goals (e.g., envisioned pedagogical strategies or pedagogical techniques). This transformation represents the first step from *information* to *knowledge*, therefore promoting a higher level of semantics.

Semantic generation of Primary content. Concept attributes, as has been mentioned, can be grouped into types. A semantically relevant subset of these types can be used to determine a semantic filter for the selection of the items that will appear in the Goal and Constraints Model. The filter represents the constraints in the GM model, while the semantics of the filter represents the goal.

For instance, a lesson dedicated to beginners can form a filter containing domain attributes of types such as: 'Introduction', 'Explanation'. These attributes can be semantically grouped in the GM as *alternative contents* (OR) and obligatory contents (AND) concepts.

Semantic Generation of Links. Links in the domain layer can be, as previously noted, hierarchical, or of another nature. These link types can be used to generate specific links at the level of the GM model.

For instance, the GM model can be generated by filtering only links of a specific semantically relevant type (e.g., only hierarchical links). These links then are semantically interpreted, therefore becoming *prerequisite* relations. In MOT, automatic transformations of hierarchical links are used to create a hierarchical, ordered link structure; i.e., the selected attribute subset will keep the same *hierarchical structure* as its DM source. However, the semantics changes from an *inclusion hierarchy* to that of a *prerequisite hierarchy*.

From Domain - to Adaptation Model (DM→AM)

The role of the adaptation model is to interpret the other models: the domain –, goal – and even presentation model. Moreover, it can update these models and generate the presentation. Typical elements of the adaptation model are condition-action (or IF-THEN) rules that change learner model variable values or presentation aspects. LAOS actually uses the LAG model (Layered Adaptive Granulation, Cristea & Calvi, 2003) to express adaptation with richer semantics. LAG has, at the lowest level, *adaptation assembly rules* such as IF-THEN rules, but wraps them in a second layer of an *adaptation language*, and at the highest level *adaptation strategies*. There are not many semantic descriptions at the lowest LAG level, hence the semantics are built into the other layers. The semantics of the adaptation language correspond to typical educational adaptation constructs that commonly appear during different adaptive interactions with the learner. The highest level, adaptation strategies, correspond semantically to pedagogic strategies.

Automatic (adaptive, adaptable) adaptation model enrichment based on the Domain Model is also a matter of semantic interpretation, with respect to a goal, e.g., a pedagogical strategy.

Automatic Semantic Rule Generation based on attribute types. Attribute types can be used to semantically create rules that control the display of specific types of attributes under specific conditions. These conditions can be automatically deduced by the system (as in adaptivity) or triggered by the AH user (adaptability).

For instance, a generated *specific* automatic adaptive rule can express the fact that we only want to show the domain attribute of type 'text' of concept *c1* after the attributes with types 'title' and 'introduction' were accessed:

IF(c1.title.access='TRUE' AND c1.introduction.access='TRUE')

THEN c1. text.available='TRUE';

Note that we wrote the condition in this form for simplification purposes, and that attribute states such as 'access' and 'available' are part of the user model.

In order for this to be a *generic* automatic transformation rule, that can be applied to *any* concept in the domain model, the rule becomes:

IF(concept.title.access='TRUE' AND concept.introduction.access='TRUE')

THEN concept. text.available='TRUE';

From Goal and Constraints - to Adaptation Model (GM→AM)

The Adaptation Model should actually work together with the Goal and Constraints Model, as the latter is the filtered version of the initial information, tailored for the group (stereotype) of learners envisioned. The Adaptation Model fine-tunes this stereotyping, catering for the individual learner's needs, as opposed to the groups needs. Enriching or generating the Adaptation Model based on the GM means semantically interpreting the GM according to an adaptation strategy or technique (e.g., based on a pedagogical strategy or technique).

Automatic Semantic Rule Generation based on Link Type. The GM, as said, contains pre-ordered and pre-selected information from the DM. This structure can already be semantically interpreted in terms of the adaptation that is to be performed on it. For instance, the GM allows 'AND' relations between concepts, as well as 'OR' relations with some weights.

These can be used to automatically generate rules that express the requirement that all concepts in an 'AND' relation must be read:

```
IF ((c.name.access='TRUE' OR c.contents.access='TRUE')
AND link(c,c2,'AND',*))
THEN { c2.name.accessible='TRUE'; c2.contents.accessible='TRUE';}
```

In a similar way, an 'OR' relationship can be semantically interpreted into inhibition rules:

IF ((c.name.access='TRUE' OR c.contents.access ='TRUE')

AND link(c,c2,'OR',*))

THEN { c2.name.accessible='no'; c2.contents.accessible='no';}

In such a way, various constructs can be automatically added to the generic adaptation rules, directly by interpreting the *goal and constraints model*.

From User - to Adaptation Model (UM→AM)

The LAOS user model is a *hybrid model* (similar to Zakaria & Brailsford, 2002). This means that the learner model consist of a *stereotype model* and an *overlay model*.

The first consists of *variable-value pairs*, which specify, information on a student; for instance:

- Interests (e.g. main interests, cross domain interests, etc.)
- current educational status
- residential constraints (e.g. preferred cities, max distance to travel per day, etc.)
- preferred study duration
- language (e.g. mother tongue, preferred study language)
- medical status
- age

Variables as above can enter conditions in adaptive rules or can be modified by these rules.

The second model specifies not only variable-values, but also the relationships between these variables, which can be deduced from the underlying domain model (or goal and constraints model).

Automatic Semantic Rule Generation based on Attribute Type. To illustrate a semantic interpretation of user model elements to generate an AM rule, we consider the state of 'interest' a learner manifests about a concept. A possible semantical interpretation of this state, evaluated via the domain overlay attribute with the same name, is to generate a rule that displays everything in the concept, if this concept is of interest to the user:

IF (concept.interest > threshold)
THEN {concept.name.available='TRUE'; concept.contents.available='TRUE';}

Note that this rule is a generic rule, which can be applied on all concepts in a concept map, drastically reducing the workload.

UM→AM: by Link Type. Link type can only be used when the UM is itself a concept map. Via UM links we can express for instance the fact that two states in the user model are related.

Here, however, we try to look at a different type of link between UM concepts. For this, let's consider the link of type 'influence'. Such a link can be automatically interpreted into a rule saying that the interest in a subject c might decrease if the user is interested in another subject c2.

```
IF LINK(c,c2,'influence',*)
THEN {c.interest= c.interest – c2.interest;}
```

CONVERSIONS BETWEEN AEH SYSTEMS

Paradigm Shift (one-to-one → many-to-many)

LAOS addresses many of the issues regarding the complexity of authoring, but cannot cover all of them. This is because the problem is compounded when one considers other factors, such as, software rot and the multitude of systems available. Software rot occurs over time because software is not maintained or software necessary for the correct workings of a program is altered in such a way that the original code ceases to function correctly. Imagine a situation where an author goes to the not inconsiderable time and trouble to create a lesson in an AEH system, be it based on LAOS or not. What happens if this system ceases to be maintained? As many AEH systems are currently developed by individual research groups around the world, the above situation has occurred many times and will occur again. Before this happens the author must consider the future. Does he stay with the old system that will slowly rot away? Or does he spend the time and effort to learn how to author in a new AEH system? It is possible that the original content is locked into the previous format, hence he may have to re-create all of his old lessons.

With the ongoing growth and maturation of AEH, these issues are raised. It is more widely recognised as desirable to move away from a "one-to-one" AH authoring paradigm to a general "many-to-many". That is, an author may create a lesson in any system he is an expert in, or wishes to spend the time and effort to learn, and export (or *convert*) this data for use in an other system. It would be of no concern if an individual system ‘rotted’ or was no longer available; a simple conversion from that system to a new one would solve this problem. This is an extended form of authoring.

What follows is a description of the first steps taken in this direction. However, rather than the ultimate goal of a “many-to-many” system we describe a half way point, a “one-to-many” methodology. Using MOT as an authoring tool (as it is based on the powerful and flexible LAOS framework) it is possible to create whatever content is desired. It is then possible to transform the lesson into one of three different formats: AHA!, Blackboard and WHURLE (actually there are *four* formats, but the original MOT delivery format requires no conversion). The process involved in doing this is described in the following sections.

Existing Multi-System Authoring Environments

In the following sections we will analyze some inter-system authoring experiments. We will discuss how learning material can be created in one system, MOT, and converted into other delivery systems. The conversions described below represent the one-to-many paradigm shift.

MOT → AHA!

MOT (My Online Teacher) is an AEH authoring system based on the LAOS framework. At the time of the writing, MOT implements the:

- domain model, as a *conceptual domain model* for courses,
- goal and constraints model, as a *Lesson map*,
- user model, featuring *stereotypes* and *overlay user model* (Wu, 2002).
- adaptation model (MOT-adapt), in the form of an (*instructional) adaptive strategy* (Cristea, 2004c) creation tool, based on an *adaptive language* (Cristea & Calvi, 2003) that uses as an intermediate representation level of LAG (Layers of Adaptive Granulation) grammar (Cristea & Verschoor, 2004)
- presentation model is currently being implemented, in the form of a hybrid model, similar to the user model.

AHA! is a general-purpose adaptive Web-based engine, first created as a simple support engine for adaptive on-line courses (De Bra & Calvi, 1998). The key features of AHA! (version 2.0) are:

- Open Source project
- Web-based adaptive engine
- Built on Java Servlet technology
- Authoring through Java Applets
- General-purpose user-model and adaptation rules
- Extensive use of XML
- Database Support using mySQL.

In addition to these, AHA! version 3.0 contains constructs called 'objects'. These 'objects' allow a complex inclusion structure of elements of a learner presentation, in a more flexible way than in earlier versions.

As MOT's first version implemented only the *domain* and the *lesson* maps, the first MOT to AHA! conversion focussed on the conversion of these maps only.

The version currently under implementation is aiming to make use of the new facilities in AHA! 3.0 and the extensions in MOT.

In the following, these conversions will be sketched separately, from a semantic and implementation point of view.

Semantic Mapping of the Domain Model

The MOT domain model layer contains a hierarchy of domain concepts and their respective domain attributes. Moreover, the DM contains also (typed and weighted) relatedness relations between domain concepts. The conversion from MOT to AHA! was initially performed using AHA! v2.0.

AHA! v2.0 only knew how to handle conditional inclusion of fragments, which are parts of an (xhtml) page. Therefore, the (xhtml) pages had to be generated for the AHA! v2.0 engine.

Note however that if a concept attribute appeared in more than one condition, it had to be pasted as a conditional fragment in each of the (xhtml) pages in which it could appear. The object inclusion in AHA! 3.0 solves this redundancy problem. This is the reason why the conversion process only started to function closer to the desired requirements with the advent of AHA! 3.0.

A conversion of MOT domain concept maps into AHA! 3.0 implies the following steps:

1. a first step of creating an XHTML (basic) resource file for every domain attribute in MOT[3]. This will generate AHA! object concepts[4] for each attribute (Attr 1 to Attr k) in Figure 2.
2. a second step of grouping of domain attributes (representing the different aspects of a concept that should appear when certain instructional strategies are triggered) into XHTML files, containing lists of 'objects', pointing to the XHTML files created in step 1. This will generate AHA! page concepts (as shown in Figure 2).
3. The actual conditions that determine which (or how many) of the alternatives are really shown to the student are written in AHA! rules during conversion

Semantically, this means that MOT domain attributes correspond to AHA! resources, whereas MOT domain concepts correspond to a special type of AHA! concept called 'page concept'.

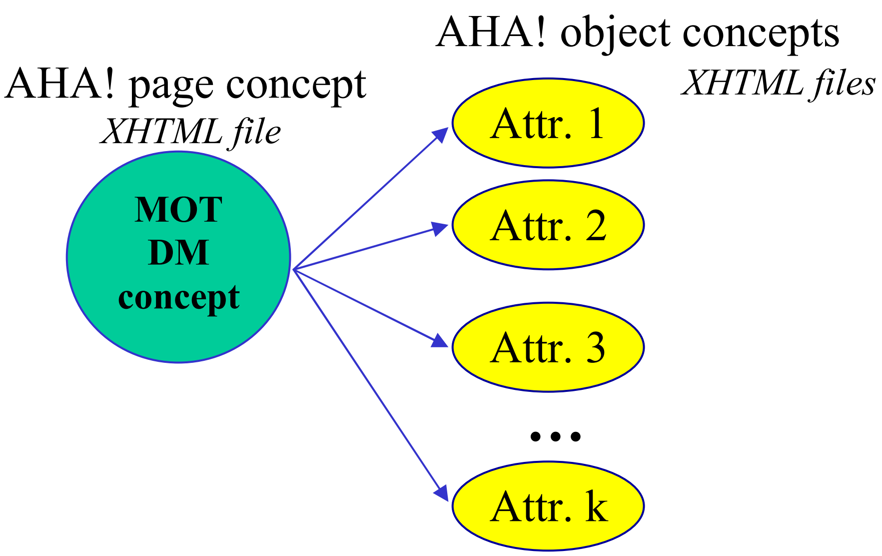

Figure 2. Semantic representation of MOT domain concepts and domain attributes in AHA!

---

[3] This only means adding a header and a footer to the attribute and saving it into a file with unique name, <file-name>.xhtml.

[4] AHA! has different types of concepts, such as object concepts, page concepts, etc. The type of the concept specifies how the AHA! delivery engine will render the content of the respective concept.

The actual representation of the domain map conversion in the AHA! implementation is shown in Figure 3. The figure shows how an AHA! page concept can be created by connecting together a list of object concepts. The actual display of the object concepts can be made to depend on some conditions (such as user preferences, state, etc.).

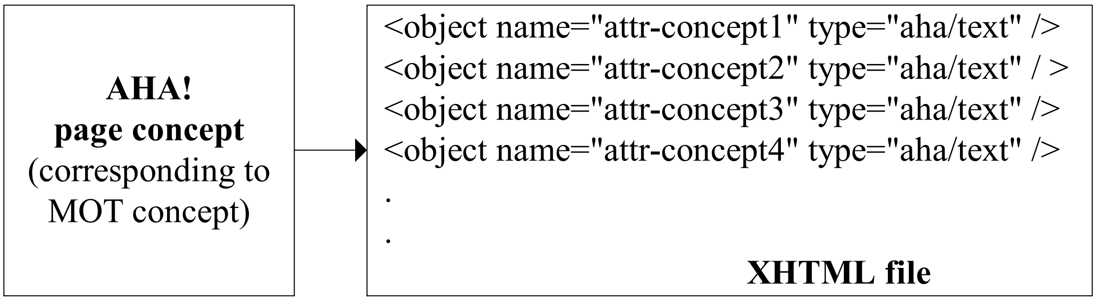

Figure 3. Implementation of MOT domain concepts and domain attributes in AHA!

Semantic mapping of the Lesson Model

The Lesson map has similar elements to the DM (as according to the LAOS model), so similar conversions can be expected. One major difference is determined by the (weighted) AND-OR relations, which can be directly interpreted as *prerequisite relations*, allowed by the AHA! engine.

Lesson map conversion into AHA! 3.0 structure is similar to the conversion of domain concept maps. The semantics are represented in Figure 4. The contents of the MOT lesson concepts have previously been created as XHTML basic resources during the domain attribute conversion, therefore this process is not repeated (i.e., the resources that are connected to the attributes Attr. 1 to 3 in Figure 4 are already bound to some AHA! object concepts, as shown in Figure 2. This is due to the LAOS restriction that attributes of the Domain map become concepts in the Lesson map).

These MOT lesson concepts are then converted into AHA! page concepts, in a similar way.

To enforce the hierarchy and order relationship, the XHTML files translating lessons contain, beside the list of object alternatives, also a separate, ordered list of sub-lesson pointers (as shown in the Figure 4).

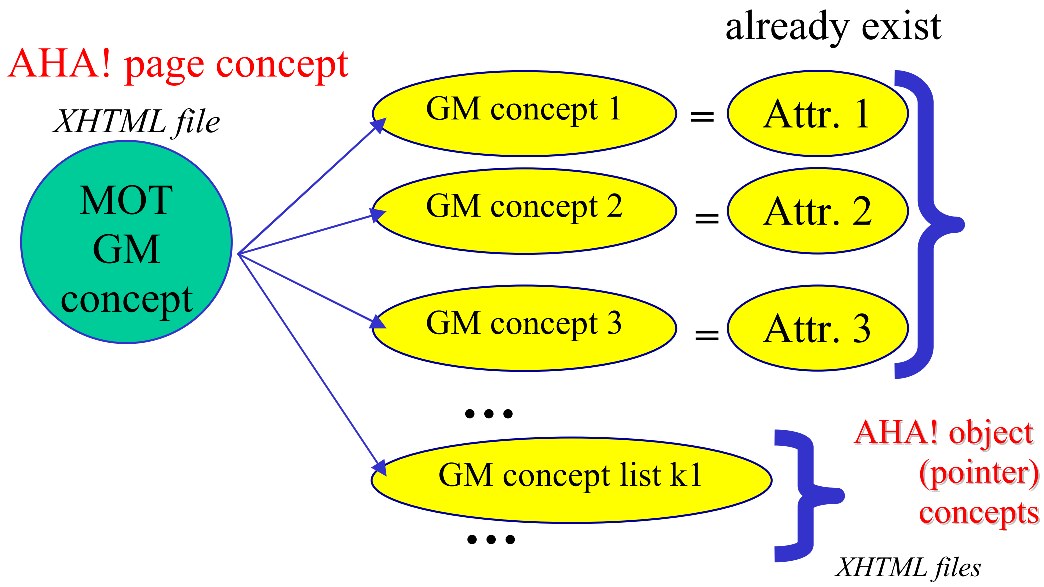

Figure 4. Semantic representation of MOT lesson concepts in AHA!

The access to sub-lessons might not be always desirable, depending on the instructional strategy. Therefore the implementation is again via the 'object' paradigm in AHA!. Moreover, a small trick is here necessary, as for sub-lessons AHA! should not display the content, but the link to the content. This can be realized in AHA! with the help of some extra link concepts containing just a link to the respective sub-lesson (see Figure 5, AHA! concept corresponding to XHTML link).

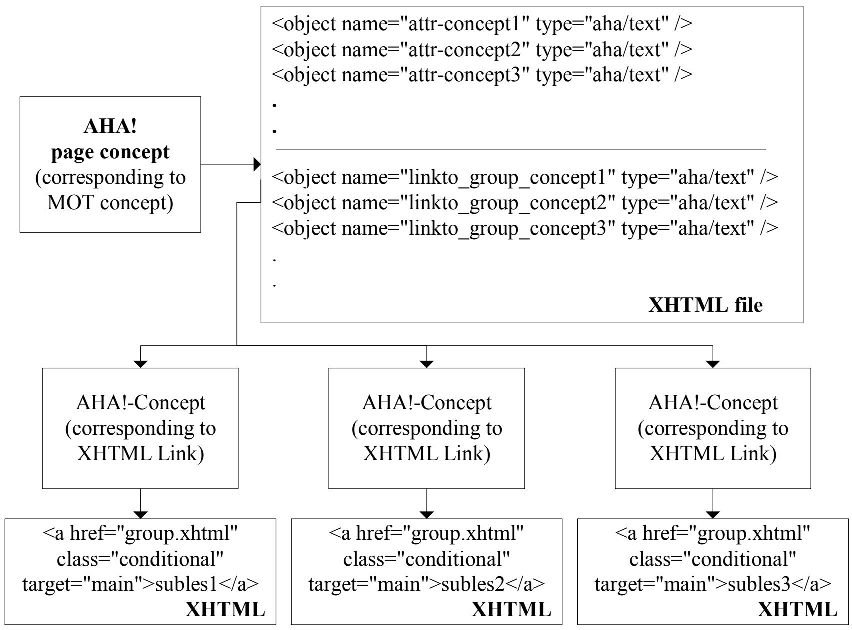

Figure 5. Implementation of MOT lesson concepts in AHA!

MOT → WHURLE

WHURLE (Web-based Hierarchical Universal Reactive Learning Environment) is an adaptive learning environment (Moore, 2001; Brailsford, 2002; Zakaria, 2003) that stores information as atomic units, called *chunks*. These are the smallest possible conceptually self-contained units of information that can be used by the system. They may be as small as a captioned image or a paragraph of text, or as large as an entire legal or historical document. Lessons consist of a collection of *chunks*, together with a default pathway, or *lesson plan*, defined by authors. The lesson plan is filtered by an *adaptation filter* that implements the user model based on data stored in the user profile.

WHURLE, however, has no specific authoring system. Both chunks and lesson plans are created using XML editors; anything from a simple text editor to an XML authoring environment may be used. Meanwhile the user profile has, in part, to be created using SQL statements which are entered manually into the MySQL database. As a novice author, with no expertise in XML or SQL, creating lessons in WHURLE is a time consuming and confusing process.

Using MOT as an authoring tool solves many of the problems that novice, or even experienced, authors have when authoring in WHURLE. Learning to use MOT is a simple feat compared to learning to author in WHURLE, as MOT authors are only required to understand HTML at the outset. There are still design decisions that need to be taken, but they are of a pedagogical nature, not a technical one.

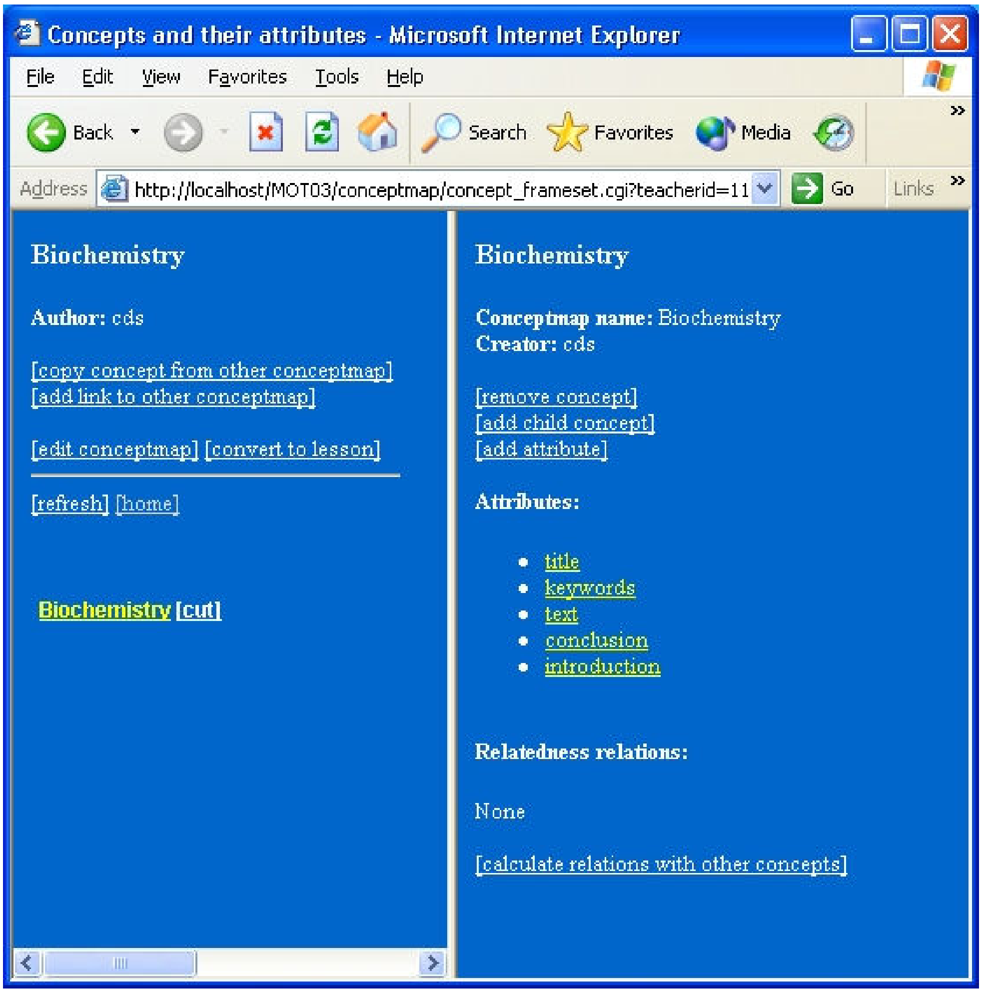

Figure 6: a MOT Domain Map, Biochemistry, with a single concept.

Authoring in MOT is a many stage process. Initially the domain maps are built; then the lesson map is created using the concepts from whichever domain maps are appropriate. The MOT-to-WHURLE conversion focuses on these two steps and therefore has two options: the conversion of domain maps or lesson maps.

```
<chunk type="" name="MOT-CM71-369.wcml"
            irks="0"
            title="Biochemistry"
            learning="ip"
            bandwidth="low"
            browsercap="low">

<versionlist>
      <version author="XXXXX" date="dddddd" />
</versionlist>

<text><p>

<p>The rest of this lesson is going to go into detail several areas of the
discpline. Areas such as:

<ul>
<li>Cellular chemistry</li>
<li>Macromolecules</li>
<li>Reaction Rates</li>
<li>Novel compunds</li>
</ul>
</p>

<p>By the end of the course you should be able to pass yours exams!</p>

<p>This course is for Degree level undergraduates.</p>
      </p></text>

      <keywords>
      </keywords>
</chunk>
```

Figure 7: The WHURLE chunk created after conversion of the MOT concept in figure 6. The original order of the attributes in the concept is maintained, hence ‘introduction’ at the end of the attribute list.

Semantic mapping of the domain model

This is a simple method of conversion resulting in a WHURLE lesson plan that has no adaptation built into it. A single MOT concept (Figure 6) is converted into a single WHURLE chunk, by gathering all of the attributes for that concept, extracting the title, keywords and placing the rest of the attribute contents into the body of the chunk.

Of course as well as chunks, WHURLE requires a lesson plan. This is also a part of the conversion output, and a section of the lesson plan produced from the domain map in figure 6 is shown in figure 8.

```
<level name="zdncrng" title="Biochemistry">
	<page>
		<chunk domain="general" stereotype1="" stereotype2="">MOT-LM71-369-0</chunk>
	</page>
</level>
```

Figure 8: a section of a WHURLE lesson plan, produced by transforming the MOT domain map in figure 6.

As a domain map has no adaptation information contained within it, it is not necessary to convert any further information. This can be seen in figure 8 where the value for the ‘domain’ is ‘general’ and both ‘stereotype1’ and ‘stereotype2’ are left blank. This indicates that there is no adaptation taking place in this lesson plan.

By ignoring adaptation, this simple form of conversion does not take full advantage of the functionality of WHURLE. For that we must turn to the second method of conversion.

Semantic mapping of the lesson model

Examine Figure 9, and the highlighted areas within it. These mark two of the major differences between a MOT domain map and a lesson map.

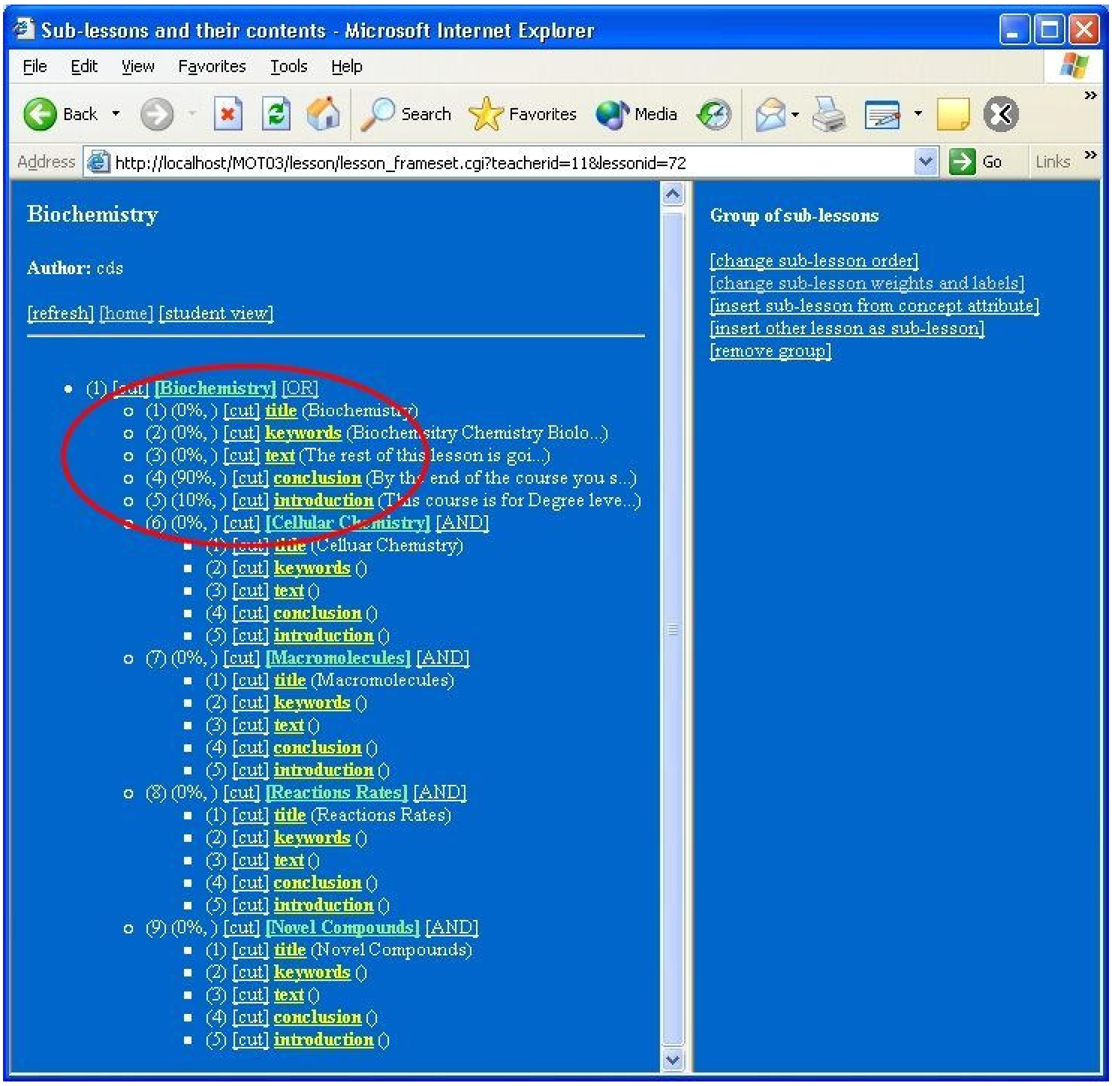

Figure 9: a MOT lesson map, the highlighted region shows that the attributes for this concept are all part of an ‘OR’ and that each attribute has a ‘weight’ – identified by the percentage (0%, 10% and 90%)

Compared to a MOT domain map, the lesson map allows for the possibility of adaptation. It does this by allowing an author to assign an ‘OR’ condition to any particular concept ('AND' is the default). This signifies that all of that concept's children, be they attributes or sub-concepts, can have a 'weight' (and a 'label', not shown) associated with them.

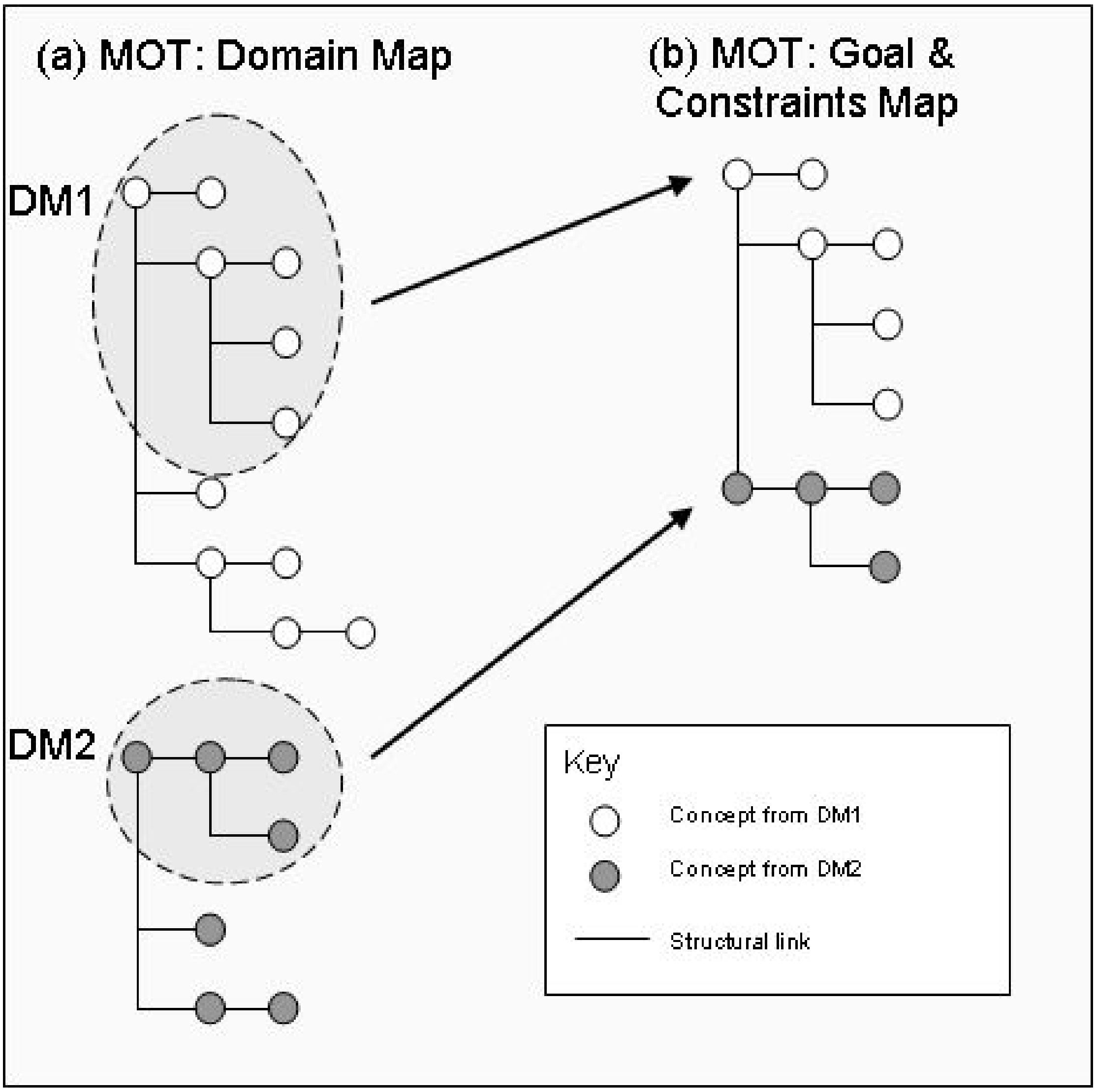

Figure 10: MOT lesson maps can be built from concepts from many domain maps.

The MOT-to-WHURLE lesson map conversion has three stages:

1) Define Structure

This stage of the conversion trawls through the lesson map and creates the WHURLE lesson plan. Each concept has to be linked to its parent, siblings and children. As lesson maps can be made up from many domain maps (figure 10), each concept must also have its parent domain map identified as this will be used as the value for the 'domain' section of a lesson plan (figure 11).

```
<page>
	<chunk domain="general" stereotype1="" stereotype2="">
		MOT-LM72-369-0
	</chunk>
	<chunk domain="119" stereotype1="beg" stereotype2="">
		MOT-LM72-369-10
	</chunk>
	<chunk domain="119" stereotype1="adv" stereotype2="">
		MOT-LM72-369-90
	</chunk>
</page>
<level name="yncioxd" title="Choose sub-topic">
	<page>
		<chunk domain="general" stereotype1="" stereotype2="">
			MOT-LM72-223-0
		</chunk>
		<chunk domain="113" stereotype1="adv" stereotype2="">
			MOT-LM72-223-90
		</chunk>
	</page>
</level>
```

Figure 11: this WHURLE page and its sub-level have been converted from a MOT lesson map using two domain maps, called '119' and '113'.

As can be seen from figure 11, the value of 'domain' is a number ('113' or '119'), because WHURLE requires a numeric value here. This is linked to the 'real' name ('MOT user guide' and 'Biochemistry' respectively) for each domain, in the WHURLE database.

Once the structure has been defined, the final process during this stage is to produce the WHURLE lesson plan. To do this, additional structures (eg XML specific data, a title, author information etc …) are added to the lesson plan.

2) Associate attributes

As the first process is ongoing (lesson plan elucidation), this second process begins: production of the chunks. Whenever a concept is encountered all of the attributes for that concept are gathered and sorted according to their MOT ‘weights’. Each weight is associated with others of the same weight, except for weight ‘0’. Weight ‘0’ is treated as a special case, as it allows the author to determine which attributes are to be ‘common’, i.e. available to all chunks created from that concept.

MOT attributes		**WHURLE chunks**		
Attribute	*Weight*	*C1*	*C2*	*C3*
Title	0	✓	✓	✓
Keywords	0	✓	✓	✓
Pattern	90		✓	
Text	10			✓
Explanation	90	No attribute contents		
Conclusion	10	No attribute contents		
Exercise	10			✓
		✓ = included in the chunk		

Table 1: a simple MOT concept with seven attributes and their associated weights will be converted into three WHURLE chunks, *C1-3*. Note attributes that are empty are ignored.

Table 1 shows which attributes will be associated with which chunks after conversion. The standard, weight '0', attributes are collected together and form chunk *C1*. Chunk *C2* is made up from the standard attributes plus all those with a weight of '90', whilst chunk *C3* collects together the attributes of weight '10' and '0'.

So far this is of no obvious use to the author. However, used in conjunction with a table like that shown in 2, it becomes possible for the author to determine his own weight boundaries when deciding which attributes belong to which WHURLE knowledge level: 'beg' (beginner), 'int' (intermediate) or 'adv'(advanced).

Weight	**Stereotype**
1 - 49	beg
50 – 89	int
90 - 99	adv

Table 2: an example weight boundary table. All MOT lesson attribute weights from 1-49 will be assigned to the WHURLE stereotype of 'beg', with 50-89 as 'int' and 90-99 as 'adv'. These boundaries are set by the author. These weight boundaries establish the WHURLE stereotype. Along with the domain for that concept they form the complete definition of the lesson plan chunk. With this done the attributes are all associated and used to produce the chunk. Note that the 'title' and 'keywords' attributes are always of weight '0' as they are required in every chunk. The chunk structure itself is similar to that displayed in figure 2.

3) Update WHURLE database

The final step in the MOT->WHURLE conversion is to create the SQL commands that will update the WHURLE database. Like MOT, WHURLE uses a MySQL database, which is used to record certain information about each of the WHURLE lessons (such as the name and unique ID of each lesson, a list of the knowledge domains (appearing as numbers in Figure 11: "<chunk domain="113" …") used in each lesson, pre-test and post-test to be used when a student first accesses a lesson, etc.).

Therefore this final step has to initially check the WHURLE database to determine what information already exists – e.g., *has a lesson of the same name already been created?* and *what Lesson and Domain IDs are available for use?* As an example: imagine that a lesson on Chemistry already exists, under the name "Chemistry", the lesson ID of "8" and the domain ID of "100". A new author has used MOT to create second lesson Biochemistry – using the domains of Chemistry, Biology and Biochemistry. The conversion system must check the WHURLE database to see if any of the domains used in the new lesson already exists. In our example, Chemistry already does, so WHURLE returns the domain ID (100) and uses this in the creation of the WHURLE Lesson Plan. Domains that are not already extant are then created, given an ID and then used in the WHURLE Lesson Plan. After this the database will be updated with all of this new information: the lesson name of "Biochemistry"; the lesson id of "9", and two additional domain IDs of "101" and "102".

The actual SQL commands are irrelevant here, as the MOT-to-WHURLE conversion program handles all of this transparently. Once the author supplies the location of the WHURLE database, everything else is automatic.

MOT → Blackboard

The ‘Academic Suite’ by Blackboard (Blackboard, 2004), contains the Blackboard Learning System. This product is not an AEH, as Blackboard has no adaptability functionality built into it. Blackboard does, however use an open architecture, which means that it is possible to extend its functionality by designing and building a Blackboard plug-in.

Even with no adaptation plug-in currently available for Blackboard, there are ways to simulate adaptation with the use of ‘adapted’ lessons. Blackboard supports the IMS Simple Sequencing Specification (SSS, 2004) which can be used to describe how learning materials can be sequenced into a specific lesson. Thus it is possible to produced ‘pre-adapted’ lessons, one for each type of user, giving the learner the illusion of adaptation.

Work has recently begun at the University of Southampton, on this innovative method for delivering adapted content in a non-adaptive system, using MOT as the authoring tool to describe the lesson before adaptation. The MOT to Blackboard conversion uses only the MOT lesson map, and like the MOT-to-WHURLE conversion it uses MOT weights to determine which attributes are delivered to which adapted Blackboard lesson. Figure 12 shows a MOT lesson map, similar to figure 9, with weights ascribed to all of the attributes.

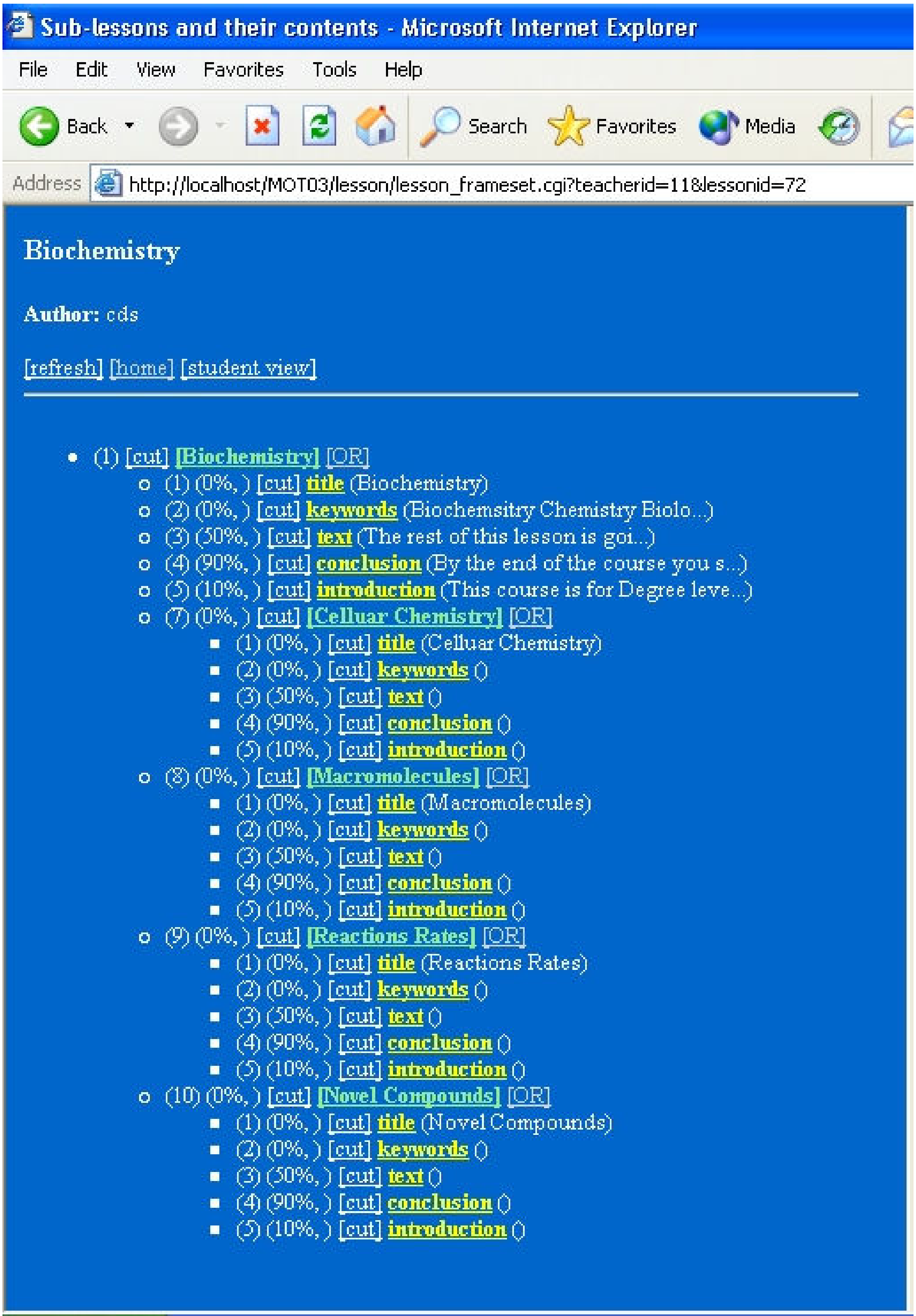

Figure 12: a MOT lesson map with each attribute given a weight.

The weights are used to determine which attributes are gathered together in to a single Blackboard lesson. Table 3 gives some example boundaries.

Weight	Learner's goal
90 +	Top of the class
70 - 90	'A' grade
60 - 70	'B' grade
50 – 60	'C' grade
Less than 50	Pass only

Table 3: some example boundaries for MOT weights, associated with a specific pass grade that a learner is aiming to achieve.

Unlike the MOT-to-WHURLE weight boundaries, where the aim to is split the content up in to ability levels (so that, for example, a beginner will only get material appropriate to a beginner's ability), the MOT to Blackboard weight boundaries are designed to create lessons appropriate to the established *goal* of an individual learner. For example, a learner can state that they only want to pass a specific subject and therefore they will only be presented with a lesson designed for 'Pass only' learners – with MOT attributes of a weight of less than 50. Figure 13 shows how a single MOT lesson can produce all of the relevant Blackboard lessons.

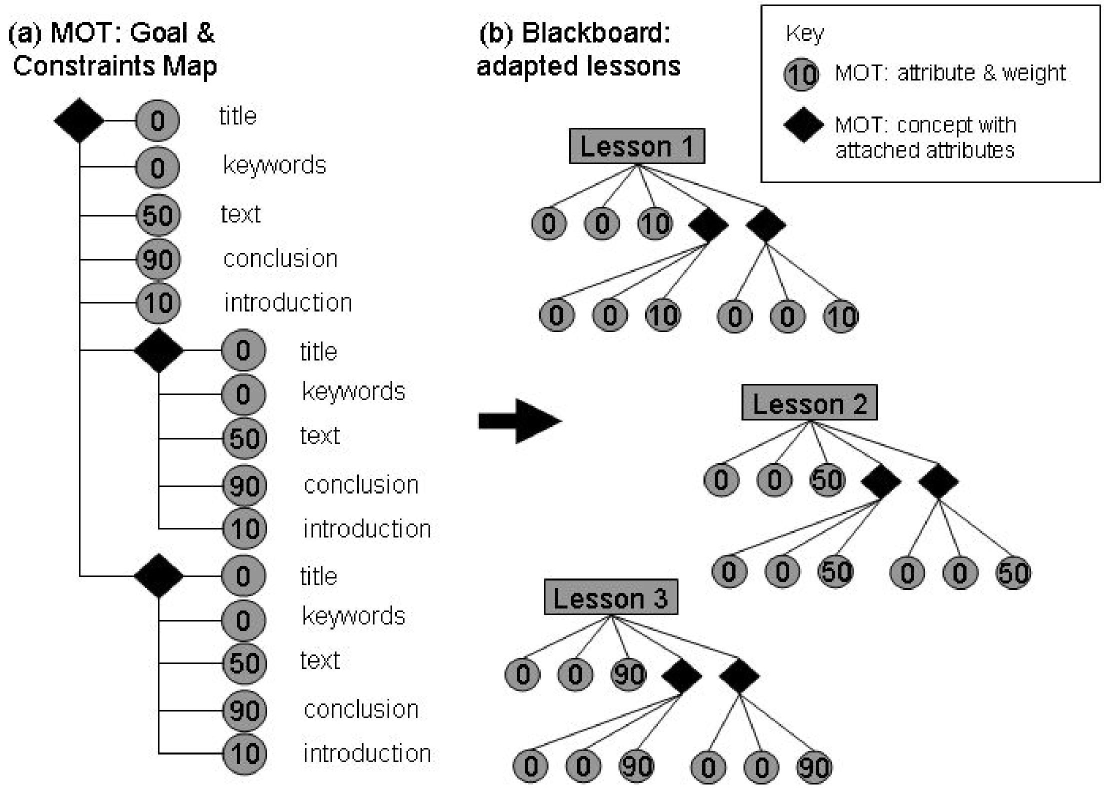

Figure 13: (a) a single MOT lesson map (goal & constraints map) would be converted to three (b) Blackboard lessons. The weights are associated into different lessons according to the weight boundaries set in table 3.

Blackboard is not an adaptive system. However it *is* in widespread use, and along with WebCT, it is one of the most popular Learning Systems in use. This widespread usage, along with its open architecture, means that designing any form of adaptation for users of Blackboard (be it illusory, via 'adapted' lessons, or by creating an adaptation plug-in) will be an important tool to advance the use of, and awareness of, AEH outside of the discipline.

The Future: Middleware

All three example conversion systems are limited in scope to the initial use of a single authoring system, MOT. No matter how good MOT is as an authoring system this is still a far step from our stated aim of a "many-to-many" paradigm for both authoring and delivery. However these first steps in that direction are vital. Firstly, a modular approach to authoring is important to encourage other AEH systems to use MOT as an authoring system; this content can be subsequently used in other systems, via conversion. Even more useful is the experience and insight that writing these modules gives to the developers. It is from these insights that a more powerful system will emerge. That system is for the future, but a brief outline of it can now be envisioned.

LAOS – LAG

As described in the LAOS Layered Model, the methodology includes an Adaptation Model layer. This layer is actually a more complex layer, as it is the one that determines the whole dynamics of the adaptive hypermedia system. Traditionally, this layer is composed of IF-THEN rules. LAG is, as said, an extension of the Adaptation Model, and its implementation is MOT-adapt.

In MOT-adapt, by using the general rule set of IF-THEN rules, it is possible for authors to write their own adaptation rules. Whilst at first glance this may sound rather complex and daunting, the LAG structure itself offers the solution to this. LAG, as previously discussed, has itself three layers of rule definition. To recapitulate, they are as follows. The first, and most basic, is the aforementioned "IF-THEN rules". The second is that of an "adaptation language", made up of more complex programming procedures. The third layer is that of broad "adaptation strategies". These are pre-written strategies that an author can use, for example, to automatically adapt all of a lesson's content depending on whether the learner is 'textual' based or 'visual/image' based.

By using LAG structure to define the conversion of adaptive behaviour of courseware, it is possible to perform flexible interpretations of the semantics of the adaptation conversion, instead of using fixed semantics.

As the model includes high-level strategies, it is possible to see the real benefits in using LAG to guide a generic authoring system. Pedagogic experts can write MOT-adapt strategies, which can then be shared with all MOT users. The author is not required to develop strategies of his own, but can always alter a pre-written strategy to suit his own specific requirements. Therefore building LAG into any future conversion system is vital.

Middleware

What is 'middleware'? Consider a heterogeneous world of AEH systems. There will be entrenched, fully developed, systems that are in use in many locations with many lessons. Then there will be new systems, still under development, informed by current research. And, of course there will be many systems between these two extremes. To have a *true* "many-to-many" paradigm, it should be possible to use any system to author for any other system. To put it another way, it should be possible to convert between any two systems of choice. MOT would no longer be the only inter-system authoring tool, and lessons written in WHURLE using an XML editor could be converted to a format that MOT can use.

How is this conversion performed? Through a piece of software that sits between each system, in the ‘middle’. This middleware would accept all conversion calls from a system and output the desired lesson(s) to the specific target system. Obviously, for such a system to function, all AEHs would have to know how to communicate with it. All data would have to have certain semantics attached to them, i.e. each would have to have some ‘meaning’ defined. Also each AEH would have to declare to the middleware what sort of data it can accept. For example it may be that WHURLE will be offered content from another AEH that will adapt around both a learner’s knowledge level (stereotype) and their language preference – WHURLE would have to declare that it would accept adaptation based on knowledge level but reject the adaptation based on language as it cannot use that.

Using a middleware system that implemented LAG would offer a great deal of power and flexibility, both to the authors of a lesson and to the learners themselves. Authors would still have to learn how to use *a* system but they would then be able to chose the simplest system appropriate to their needs and have the content delivered to any AEH anywhere in the world.

DISCUSSION AND CONCLUSION

Adaptive Educational Hypermedia (AEH) aims to deliver flexible and appropriate educational materials to each student. This is in response to the inflexible and inappropriate use of learning resources in many static online Educational Hypermedia systems.

Authoring adaptive materials is no simple matter. An author must determine which of a multitude of AEH systems best suit his desires and requirements; this can involve a great deal of research. Even once a specific AEH is chosen, it may not be the correct one: it may no longer be supported, or it may lock down the content in a format that the author does not wish.

Moving on from these initial problems an author comes face-to-face with the many difficulties involved in actually producing adaptive content. For example the multiple versions of information required for each type of learner, each possible adaptation of the content. With each problem the author has to develop a new solution, but a solution that is limited to that specific AEH. Whilst some of the expertise gained in writing lessons can be applied to multiple AEH systems, if an author wishes to move onto a new system then much of the hard earned expertise becomes worthless.

After outlining the difficulty of the authoring task for Adaptive Hypermedia, we proposed solutions in the form of automatic authoring techniques, achieved by semantic interpretation of partial content of AH, as in internal transformations, or in semantic interpretation of integral AH content, as in the external conversions into several AH delivery platforms.

Internal transformations within an AEH authoring system allow the *author to write only a minimal amount of material*, which will be exploited and semantically interpreted automatically by the system into a complete AEH.

Moreover, this chapter has introduced a solution to this perennial authoring problem; that of an *author creating educational content once* (in a generic authoring environment, such as MOT) and subsequently being able to view it in multiple AEH delivery environments. This “write once, use many” approach is of course only an intermediate step towards a middleware system that will allow a dynamic interchange of information between all AEHs.

This ‘inter-operability’ between AEH systems has recently been identified by the community as being important. For example, AHA! (De Bra & Calvi, 1998), a well known AEH system in academic circles, has also experimented with conversion; notably, authoring with Interbook for AHA! (De Bra et al., 2003), and using AHA! as the user model server for Claroline (Arteaga et al., 2004).
Both of these developments represent a step in the right direction, and demonstrate the fundamental principle of AEHs being able to interchange data. However, they both lack the co-ordination represented by the three examples given in this chapter, as they are both separate developments that do not reference a common interface system, such as LAOS. Due to the fact that these conversions were both uniquely designed to interface with AHA! and no other AEH, they do not really move closer to a “many-to-many” approach.

By examining the conversion between MOT and WHURLE in detail, we can perceive a great many conceptual similarities. WHURLE is organised by lesson plans and the pages within them, which are clearly equivalent to MOT lesson maps and concepts respectively. As the two systems developed independently this similarity probably grew out of the comparable aims of each system, a case of parallel evolution.

Even systems which are conceptually much more divergent than MOT and WHURLE, such as MOT compared to Blackboard, are nevertheless similar enough to allow for a generic conversion system to deliver an illusion of adaptation.

Generic conclusions drawn from a few test cases such as these should be treated with caution. However, within the discipline of AEH it could be productive to consider the conclusions that can be drawn from the insights gained during these conversions. The obvious conclusion is that many AEH systems will share a similar semantic structure, or that, at the least, there will be enough of an overlap between the semantics of each system to allow for a productive conversion to occur.

This overlap must be made when preparing to convert between two systems, semantic mapping of the educational materials and the system data models is vital. Without such a mapping it is impossible to state that a 'title' attribute in MOT is used in the same manner as a 'title' section in the target AEH. Without such an assurance it is impossible to be certain that any conversion system will in fact produce output which retains the same meaning.

Using a layered framework such as LAOS has another advantage for authors in addition to those already discussed, as LAOS has its own semantics that are built into each layer. The author need no longer consider the semantics of the material he is creating, as this will automatically be assigned when he designs the lesson. From the point of view of an AEH developer, we claim that, if a target AEH system implements LAOS then the target semantics are already known and a conversion module is straightforward to create.

ACKNOWLEDGEMENTS

This research is linked to the European Community Socrates Minerva project ADAPT: "Adaptivity and adaptability in ODL based on ICT" (project reference number 101144-CP-1-2002-NL-MINERVA-MPP).

REFERENCES

ADAPT EC project, http://wwwis.win.tue.nl/~acristea/HTML/Minerva/index.html

Berners-Lee, T., Semantic Web Status and Direction ISWC'03 keynote, ISWC'03, http://www.w3.org/2003/Talks/1023-iswc-tbl/slide26-0.html

Arteaga, C., Fabregat, R., Eyzaguirre, J. & Merida, D. (2004) Adaptive Support for Collaborative and Individual Learning (ASCIL): Integrating AHA! and CLAROLINE, Adaptive Hypermedia and Adaptive Web-based Systems, Paul De Bra & Wolfgang Nejdl (Eds), LNCS 3137, Springer, 279-282.

Blackboard, Blackboard Academic Suite, http://www.blackboard.com/products /academic/index.htm, 2004

Brailsford, T.J.; Stewart, C.D.; Zakaria, M.R. & Moore, A. (2002). Autonavigation, Links and Narrative in an Adaptive Web-Based Integrated Learning Environment.11th International World Wide Web Conference, Honolulu, Hawaii, 7-11 May 2002.

Brusilovsky, P. (2001a). Adaptive hypermedia, User Modelling and User Adapted Interaction. Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110.

Brusilovsky, P. (2001b). Adaptive Educational Hypermedia (Invited talk). 10th International PEG conference, Tampere, Finland, June 23-26, 8-12.

Brusilovsky, P., Eklund, J., and Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems (Proceedings of Seventh International World Wide Web Conference, 14-18 April) 30 (1-7), 291-300.

Carro, R.M., Pulido, E. & Rodríguez, P. (2001). TANGOW: a Model for Internet Based Learning. International Journal of Continuing Engineering Education and Life-Long Learning, IJCEELL, Pub. UNESCO. Special Issue on "Internet based learning and the future of education",

http://www.inderscience.com/ejournal/c/ijceell/ijceell2001/ijceell2001v11n12.html

Coffield, F. (2004) Learning Styles and Pedagogy in post-16 learning: A systematic and critical review. Learning & Skills research centre.

http://www.lsda.org.uk/files/pdf/1543.pdf

Cristea, A. (2004). Flexibility of Automatic Authoring for the Semantic Web, WWW'04, Workshop on Application Design, Development and Implementation Issues in the Semantic Web, May 18.

Cristea, A. & Cristea, P. (2004) Evaluation of Adaptive Hypermedia Authoring Patterns During a Socrates Programme Class, Advanced Technology for Learning Journal, ACTA Press, 1(2), 115-124,

http://www.actapress.com/journals/onlinejournals.htm

Cristea, A.I. (2003). Automatic Authoring in the LAOS AHS Authoring Model. Hypertext'03. Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems.

Cristea, A.I., & Aroyo (2002). L. Adaptive Authoring of Adaptive Educational Hypermedia, AH'02. Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, Springer, 122-132.

Cristea, A.I., & Calvi, L. (2003). The three Layers of Adaptation Granularity. UM'03. Springer.

Cristea, A.I., & De Bra, P. (2002). Towards Adaptable and Adaptive ODL Environments. AACE E-Learn'02 (Montreal, Canada, October 2002), 232-239.

Cristea, A.I., & Kinshuk. (2003). Considerations on LAOS, LAG and their Integration in MOT. ED-MEDIA'03.

Cristea, A. & De Mooij, A. (2003a). Evaluation of MOT, an AHS Authoring Tool: URD Checklist and a special evaluation class. CATE'03 (International Conference on Computers and Advanced Technology in Education) Rhodos, Greece, IASTED, ACTA Press, ISBN 0-88986-361-X, pp. 241-246.

Cristea, A. & De Mooij, A. (2003b). Designer Adaptation in Adaptive Hypermedia. ITCC'03 (Las Vegas, US 28-30 April) IEEE Computer Society.

Cristea, A. & De Mooij, A. (2003c). LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. WWW'03, Alternate Education track. (Budapest, Hungary 20-24 May). ACM.

Cristea, A. & De Mooij, A. (2003d) Adaptive Course Authoring: MOT, My Online Teacher. ICT-2003, IEEE LTTF International Conference on Telecommunications, "Telecommunications + Education" Workshop (Feb 23 - March 1, 2003 Tahiti Island in Papetee - French Polynesia).

De Bra, P., Santic, T., Brusilovsky, P., (2003). AHA! meets Interbook, and more... Proceedings of the AACE ELearn 2003 Conference, Phoenix, Arizona, November 2003, pp. 57-64.

De Bra, P. & Calvi, L. (1998). AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, 4, Taylor Graham Publishers, 115-139.

T. R. Gruber. (1993). Toward principles for the design of ontologies used for knowledge sharing. Presented at the Padua workshop on Formal Ontology, March, to appear in an edited collection by Nicola Guarino, http://ksl-web.stanford.edu/KSL_Abstracts/KSL-93-04.html.

MOT download at: http://adaptmot.sourceforge.net/ ; try online at: http://e-learning.dsp.pub.ro/mot/; http://e-learning.dsp.pub.ro/motadapt/

Moore, A., Brailsford T. J. & Stewart, C. D. (2001) Personally tailored teaching in WHURLE using conditional transclusion. Proceedings of the twelfth ACM conference on Hypertext and Hypermedia, Denmark.

Murray, Tom. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education, 10, 98-129, http://aied.inf.ed.ac.uk/members99/archive/vol_10/murray/paper.pdf.

Semantic Web, WC3. http://www.w3.org/2001/sw/

Murray, T. (2003). "MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks." J. of Artificial Intelligence and Education, Vol. 13 (Special Issue on Adaptive and Intelligent Web-Based Systems).

Specht, M., Kravcik, M., Pesin, L. & Klemke, R. (2001). Authoring Adaptive Educational Hypermedia in WINDS, ABIS Workshop, http://www.kbs.uni-hannover.de/~henze/ABIS_Workshop2001/final/Specht_final.pdf

SSS, IMS Simple Sequencing Specification, http://www.imsglobal.org/simplesequencing/index.cfm, 2004

Stach, N., Cristea, A.I. & P. De Bra, P. (2004). Authoring of Learning Styles in Adaptive Hypermedia: Problems and Solutions, WWW'04, 13th International World Wide Web Conference, May, New York, US, 104-113.

Wu, H. (2002). A Reference Architecture for Adaptive Hypermedia Applications, doctoral thesis, Eindhoven University of Technology, The Netherlands, ISBN 90-386-0572-2.

Wu., H., Houben, G.J., De Bra, P. (1998). AHAM: A Reference Model to Support Adaptive Hypermedia Authoring, Proc. of the "Zesde Interdisciplinaire Conferentie Informatiewetenschap", pp. 77-88, Antwerp.

Zakaria, M.R.; Moore, A.; Stewart, C.D. & Brailsford, T.J. (2003). "Pluggable" user models for adaptive hypermedia in education. Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia, August 26-30, 2003, Nottingham, UK. pp 170-171.

Zakaria, M.R. & Brailsford, T.J. (2002). User Modelling and Adaptive Educational Hypermedia Frameworks for Education. *New Review of Hypermedia and Multimedia*, 8, 83-97.