Towards the Validation of Adaptive Educational Hypermedia using CAVIAr

Mark Melia and Claus Pahl

School of Computing, Dublin City University, Dublin 9, Ireland
mmelia@computing.dcu.ie

Abstract. Migrating from static courseware to Adaptive Educational Hypermedia presents significant risk to the course creator. In this paper we alleviate some of this risk by outlining how the CAVIAr courseware validation framework can be used to validate some pedagogical aspects in Adaptive Educational Hypermedia. To allow for this we present a novel method for interoperability in Adaptive Educational Hypermedia using Model Driven Engineering methodologies.

1 Introduction

The authoring of Adaptive Educational Hypermedia (AEH) is a major task for any course creator to undertake. The cost in time and effort leave many considering if the actual end product is cost-effective. Although recent advances in this area have been made, with the emergence of dedicated AEH authoring tools such as MOT [1] and the ACCT [2], there is still no way to check developed AEH for specific pedagogical problems.

Courseware validation is a design activity that automatically ensures the presence of certain structural and pedagogical characteristics in constructed courseware. Courseware validation allows the course creator to minimise the pedagogical problems which the learners must deal with when using immature courseware.

Using courseware validation in AEH, allows the course creator to automatically test the AEH for specific pedagogical problems, which may not be possible to check otherwise due to AEH's adaptive nature. This reduces the risk for the course creator, who wishes to migrate away from a static courseware and use AEH to deliver a course.

In this paper, we investigate how one AEH specification, the LAOS model can be validated using the Courseware Authoring Validation Information Architecture (CAVIAr) [3]. The paper firstly outlines the respective technologies, LAOS first and then the CAVIAr. Section 4 and 5 then introduces modeling technologies and methodologies and demonstrate how they are used to convert LAOS to CAVIAr for validation. Section 6 steps through the validation process, we conclude the paper in section 7 outlining our contribution.

2 MOT, LAOS and AEH Interoperability

The "My Online Teacher" (MOT) system [1], allows course creators to create adaptive courses using the LAOS conceptual architecture for adaptive hypermedia [4]. LAOS consists of five layered maps, where the higher layers are defined in terms of the lower layers. The layers are as follows starting with the lowest layer:

- **domain map** - "organises and structures the actual resources of the learning environment, as well as their intrinsic characteristics" [1].
- **goal and constraints map** - "this model filters, regroups and restructures the domain model, with respect to an instructional goal used to express educational goals" [1]. This is done by specifying the instructional weights of domain map concepts and by ordering the domain concepts.
- **user map** - used to specify the user knowledge, interests and learning styles.
- **adaptation map** - defines adaptive rules in terms of the lower layers. This map is defined using LAG, a 3-tier adaptive rule specification [5] .
- **presentation map** - defines course delivery environments variables, allowing the AEH to adapt to the delivery environment being used by the learner.

MOT is purely an AEH authoring environment, it does not allow for the delivery of AEH. In order for delivery of AEH material created using MOT must be delivered using an AEH delivery environments, such as AHA! [6] or WHURLE [7].

In order for the AEH developed using MOT to be delivered in an AEH delivery platform it must be interoperable with that delivery platform. To do this, Cristea et. al. makes the distinction between static and dynamic elements of the LAOS [1]. Static elements are exported from MOT through a common language, or lingua franca, known as the Common Adaptation Framework (CAF), which captures the domain map and the goal and constraint map. Dynamic elements, which describe the adaptive nature of the AEH and are captured using LAG.

MOT exports to CAF by converting the domain map and the goal and constraint map, which is stored are the MOT database, to the CAF XML specification, this can then be imported by the AEH delivery environment.

LAG captures the adaptation rules for AEH. The top level of the 3-tier LAG model is adaptation strategies, which are built on adaptation languages, which, in turn are built on direct adaption rules. In the LAOS context LAG direct adaptation rules are defined in terms of the lower layer maps. The LAG direct adaptation rules are IF-THEN or condition-action style rules, defined in a context-free BNF (Backus-Naur Form) style meta-syntax notation [1].

3 CAVIAr Courseware Validation

The CAVIAr is used in courseware authoring for automatic validation of a variety of courseware structural and pedagogical concerns including:

[1] http://wwwis.win.tue.nl/ acristea/MOT/help/LAGgrammar.doc

– inter-conceptual courseware sequencing - pedagogical concerns regarding the sequencing of concepts in courseware [8]
– intra-conceptual courseware sequencing - pedagogical concerns teaching one concept [9]
– the appropriateness of the type of learning material used at particular points in courseware
– courseware consistency
– elements of the instructional design in use in the courseware

The CAVIAr model allows for the course creator to identify instructional problems in the courseware prior to delivering it to learners. This is important as it allows the course creator to be confident that particular types of courseware problems are not present in the courseware developed. This allows formative evaluation of courseware to evaluate more complex pedagogical issues in the courseware.

Courseware validation using CAVIAr is achieved by modeling the courseware construction concerns. The CAVIAr uses a modeling structures very similar to that of LAOS, using four modeling layers as follows:

– **Domain Model** - a pedagogically neutral conceptual graph, used to structure knowledge to be covered in courseware
– **Learning Context Model** - Defines conceptual sequencing constraints and the learner stereotypes, each learner stereotype is defined as having assumed initial knowledge and a course goal in terms of the domain model
– **Courseware Model** - The courseware model is composed of two parts:
  • courseware structure, structured using courseware topics, where topics contains learning resources.
  • learning resource model, which contains a model representation of Learning Objects (LOs) and their metadata
– **Validation Model** - A constraint model which defines valid courseware

It is important to identify how the CAVIAr faclitates the representation of adaptive courseware, allowing for the mapping from AEH to CAVIAr. As we have outlined the courseware model defines the courseware structure and the LOs in the courseware. The courseware model is defined using a metamodel, an excerpt of which is in figure 1.

Adaptivity is achieved in a courseware model in two ways - specifying a "SEQUENCED_AFTER" relationship between two topics and by specifying an entryLearner requirement for a topic. The "SEQUENCED_AFTER" relationship allows the course creator to specify explicit sequencing constraints between topics. The entryLearner requirement allows the course creator to place a gate condition on a topic, so that the topic is only delivered to learners which satisfy the entryLearner requirement at any given point in time.

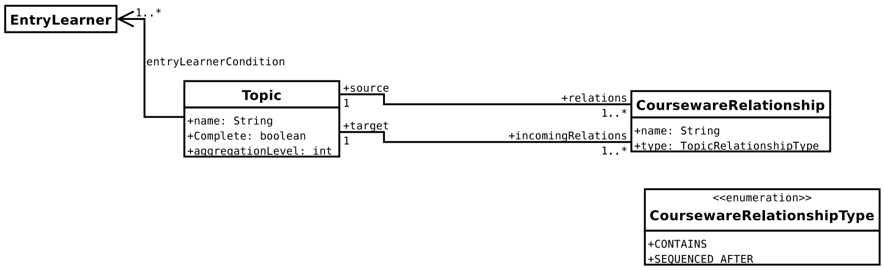

Fig. 1. CAVIAr courseware metamodel excerpt

4 Model Driven Engineering and Courseware Development

In our previous work we have outlined how Model Driven Engineering (MDE) methodologies, which are traditionally used in the development of software, can be used to develop courseware [10]. In this work the course creator defined a courseware sequence using an UML Activity Diagram, which was then transformed into a courseware specification using a model transformation language.

A metamodel defines the syntax and semantics of a model. Metamodels are defined by metametamodels. Model transformations have mapping defined at the metamodel level. Model transformations allow for the transformation of a model, which is an instance of one metamodel to a model which is an instance of a different metamodel. Figure 2, outlines model transformations defined at the metamodel level, and the actual mapping at the model level.

It should be noted that the metamodels and the model transformation definition must be defined as instances of a common metametamodel.

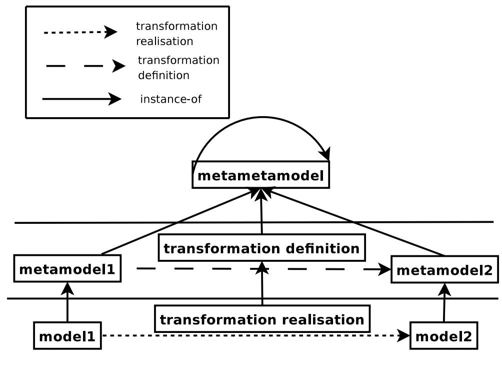

Fig. 2. Model transformations

Tool support for MDE is provided by the Eclipse Modeling Framework (EMF) [11]. We use EMF to represent CAVIAr models. EMF was used as it provided support for the following functions:

- a method for defining metamodels using ECore
- allowed for model transformations through the Atlas Transformation Language (ATL) [12]
- provided metamodel management infrastructure

5 Transforming LAOS to CAVIAr

In order to validate AEH defined by MOT using CAVIAr, the LAOS model must be used to generate a CAVIAr models. To do this we define metamodels for LAOS, one looking at LAOS static elements in CAF and the other its adaptive rules in LAG. We also define transformation from the LAOS metamodel to the CAVIAr metamodel, by identifying the relations between the metamodels.

In this section, we firstly outline the definition of CAF metamodel and its transformation relations to CAVIAr and then do the same for LAG. We note that the transformations specified here are example mappings, all model mappings can be customised by the course creator to represent their own opinions on the relationship between LAOS and CAVIAr.

5.1 CAF Transformation

To create a CAF ECore metamodel, we used the CAF XML definition, defined using a DTD [1]. This was converted to an XML schema using XMLSpy [13]. To create the CAF ECore Metamodel we converted the XML schema to an ECore model using EMF and then performed some minor alterations, as follows:

- created an explicit link between Link and Attribute
- added "value" attribute to CAF elements which contain text
- specified which relationships were ordered

The final CAF metamodel is illustrated in figure 3.

Once the CAF ECore metamodel is defined, the transformation between the CAF and CAVIAr metamodels can be defined using a model transformation language such as the Atlas Transformation Language (ATL) [12] or OMG's Query View Transformation (QVT) [14]. Here we define the transformation specifications at a high level.

Generating the CAVIAr Domain Model In order to define the CAVIAr domain model, we have defined a model transformation from the LAOS domain model concept map to the CAVIAr domain model.

In this transformation, the CAF domain model concept is related to the CAVIAr domain model concept. The conceptual composition relationship in CAF, which relates two CAF concepts together, is transformed to the CAVIAr ConceptRelationship class of type "NARROWER", where the contained concepts in LAG are narrower in scope to that of the containing concept in CAVIAr.

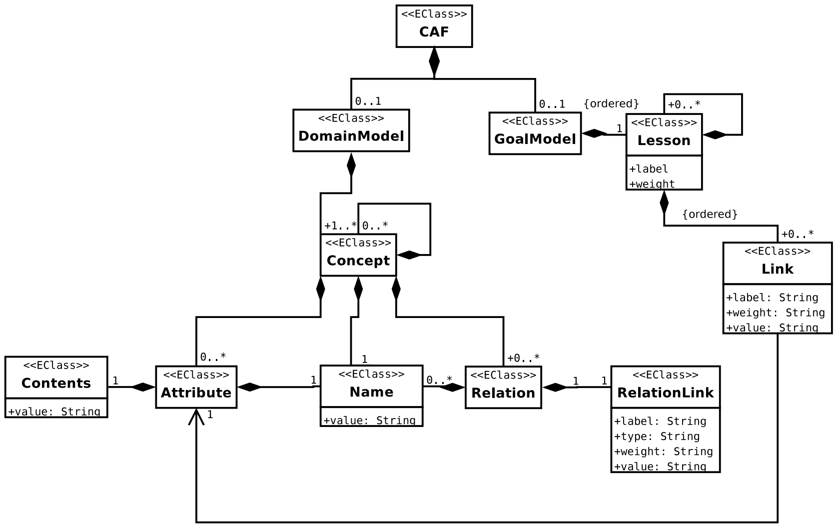

Fig. 3. CAF Metamodel defined using ECore

Generating the CAVIAr Learning Context Model The CAVIAr learning context is defined using the CAF goal and constraint model definition. Mapping is defined as follows:

- the CAF goal and constraints model is transformed into a single generic learner stereotype in CAVIAr
- CAF lesson goals are transformed into CAVIAr goals for the generic learner stereotype
- conceptual sequencing data in CAF lesson is transformed to PRE_REQUISITE relationships between concepts in CAVIAr

Generating the CAVIAr Courseware Model - Courseware Structure A courseware model is not defined by the CAF model, but can be derived using the domain model. In LAOS, the domain model contains the educational content to be delivered to the learner. We can therefore infer that each of the concepts in the domain model are also courseware topics in the courseware model.

In defining the transformation from the CAF model to the CAVIAr courseware model, we specify a 1:1 relation between the concepts in CAF and the CAVIAr courseware topics. Concepts contained in other concepts in CAF are transformed to subtopics in the CAVIAr courseware model.

Generating the CAVIAr Courseware Model - Learning Objects and Learning Object metadata In CAVIAr learning material is typically Learning Objects (LOs), and are annotated with metadata. This metadata can be used to determine the suitability of the LO at some point in the courseware. In AEH, the domain model defines what is in the AEH lesson. The domain model not only defines a conceptual structure of the AEH course but also defines the learning content. In the LAOS, the learning content is defined in concept attributes.

To generate LOs from the LAOS, we transform each conceptual attribute to a LO. The LO metadata is automatically derived for each LO generated, using the attribute type (e.g. title, conclusion) and the concept the attribute is associated with.

5.2 LAG Transformations

LAG rules are used to define adaptivity in LAOS (section 2). CAVIAr adaptation is provided by specifying restrictions on the sequencing of topics and restrictions on learner profiles which can access a topic. This type of adaptivity is defined using modeling constructs, such as defining a sequencing relationship between topics.

We wish to take the LAG adaptivity rules and transform them into CAVIAr courseware model restrictions. To do this the LAG language must be defined in the modeling technical space. We have defined a limited metamodel for the LAG abstract syntax in figure 4. This metamodel allows us to represent LAG in the modeling space by parsing a LAG rule and creating a LAG model. The LAG model can then be transformed and integrated into the CAVIAr model created using the CAF in section 5.1.

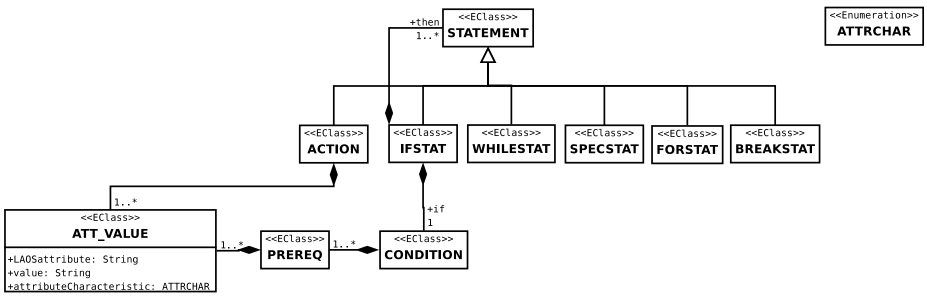

Fig. 4. LAG defined as ECore metamodel

Transformation rules can then be defined from the LAG metamodel to the CAVIAr metamodel. In the following we outline an adaptive rule which is commonly used in LAOS to define AEH, and describe the transformation definition which converts the LAG rule to the CAVIAr.

Transforming LAG Sequencing Rule LAG sequencing rules specify when a particular part of the domain model is accessed, it renders a different part of the AEH available to the learner.

An example of a LAG sequencing rule is as follows (listing 1.1):

Listing 1.1. LAG sequencing rule

```
IF (DM.Concept.title.access == 'true') THEN
   (DM.Concept.text.show == 'true')
```

The rule above states that if a domain model's concept title is accessed, then the text for that domain model concept is shown to the learner. This type of LAG rule is made up of two different parts, an IF condition and an action. The condition and action are composed by checking (condition) and then setting (action) a characteristic of a domain model concept's attribute in LAOS. The condition checks the attribute "title" for domain model concepts has been accessed - "access" being the characteristic. In turn, the action sets the LAOS "text" attribute to be shown - "show" being the characteristic being set.

This rule is parsed and creates an instance of the the LAG metamodel - a LAG model - as illustrated in figure 5.

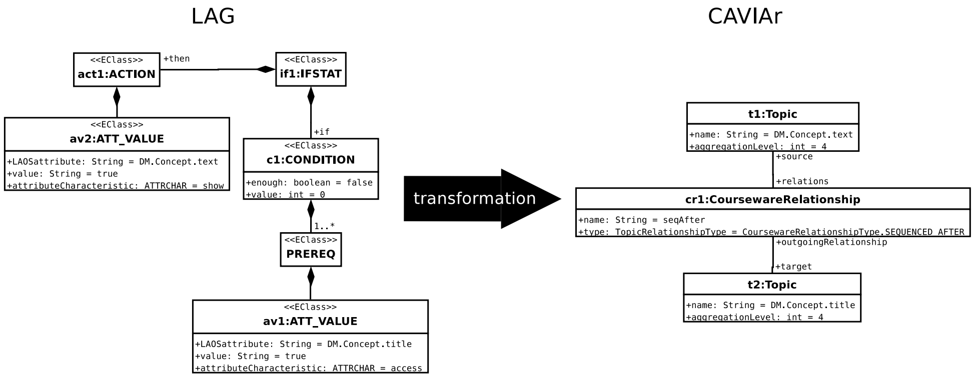

Fig. 5. Transformation of LAG model to CAVIAr courseware model

When a LAG model has been constructed for the rule in listing 1.1, the LAG rule can be transformed into the CAVIAr courseware model. To do this a transformation mapping from the LAG metamodel to the CAVIAr metamodel is defined. This transformation states when DM.Concept.title attribute is accessed show the DM.Concept.text attribute. The transformation maps this type of LAG rule to a CAVIAr courseware model where each attribute in the LAG condition and action is a courseware topic. The topic mapped to the title attribute is the source of a "SEQUENCED_AFTER" CoursewareRelationship where the target is the topic mapped to the text attribute. We have demonstrated this through an example transformation in figure 5.

6 Validating AEH using CAVIAr

In this section, we generalise the LAOS validation methodology we have presented in this paper and examine how AEH is validated in general.

When AEH is being validated for the first time, a metamodel for the AEH data models must firstly be defined. This allows the AEH to be used in the modelling technical space. The AEH native data models must be parsed to create an instance of the metamodel defined. Transformations to the CAVIAr must then be defined to map the AEH being used to the CAVIAr. The AEH metamodel and transformation to CAVIAr once defined, can be reused.

To validate AEH the course creator must then define the validation model for the CAVIAr. The validation model specifies constraints that must be adhered to in the AEH and are defined in the context of the CAVIAr models. For example, the course creator may specify that all concepts covered in the AEH are introduced with a motivating example and delivered before any other material on that concept is delivered to the learner. The course creator may feel that the AEH has been defined this way but wants to guarantee it through CAVIAr validation. The course creator defines this as a constraint on the CAVIAr model.

The validation is then run using the CAVIAr validation engine, this validates the generated models against the CAVIAr model constraints specified in the validation model. If any of the validation model constraints are breached, the course creator is notified and he or she can then rectify them in the AEH.

7 Conclusion

In this paper we have described courseware validation as a method for course creators to minimise the risk involved in creating and deploying AEH. The CAVIAr has been introduced in the AEH context, as a way for course creators to test the AEH developed for specific pedagogical concerns.

To enable interoperability between the LAOS and CAVIAr, we have outlined the application of MDE technologies and methodologies, provided model mapping from the LAOS to the CAVIAr, and detailed an implementation infrastructure with which the conversion from LAOS to CAVIAr can take place. AEH interoperability has been investigated in a number of papers, much of this work concentrates on once off conversions between two AEH technologies [7, 1, 15, 16]. In this paper we have outlined how MDE offers a generic approach to AEH interoperability, where interoperability can be achieved when a metamodel is defined for the AEH technology in use and transformations between the metamodels are implemented. The methodology outlined is also highly customisable, all the AEH to CAVIAr mappings can be changed to reflect the course creators own opinions on metamodel relationships.

References

1. Cristea, A., Smits, D., deBra, P.: Towards a generic adaptive hypermedia platform: a conversion case study. Journal of Digital Information **8**(3) (2007)

2. Dagger, D.: Personalised eLearning Development Environments. PhD thesis, University of Dublin (2006)
3. Melia, M., Pahl, C.: An information architecture for validating courseware. In Massart, D., Colin, J.N., eds.: Proceeding of the First International Workshop on Learning Object Discovery and Exchange (LODE2007) at EC-TEL2007, Crete, Greece, CEUR Workshop Proceedings (2007)
4. Cristea, A.I., de Mooij, A.: LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. In: Proceedings of The Twelfth International World Wide Web Conference (WWW03), Alternate Track on Education, ACM (2003)
5. Cristea, A.I., Verschoor, M.: The LAG Grammar for Authoring the Adaptive Web . In: Proceedings of The International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 1. (2004) 382–386
6. Cristea, A.I., Smits, D., de Bra, P.: Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia. In: Proceedings of The Third International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia at AIED05. (2003)
7. Cristea, A.I., Stewart, C., Brailsford, T., Cristea, P.: Evaluation of Interoperability of Adaptive Hypermedia Systems: testing the MOT to WHURLE conversion in a classroom setting. In: Proceedings of The Third International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia at AIED05. (2003)
8. Melia, M., Pahl, C.: Pedagogical validation of courseware. In Duval, E., Klamma, R., Wolpers, M., eds.: Proceedings of the Second European Conference on Technology Enhanced Learning. Number 4753 in LNCS, Springer (2007)
9. Ullrich, C.: Course generation based on HTN planning. In: Proceeding of the Thirteenth Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems. (2005) 74–79
10. Melia, M., Barrett, R., Pahl, C.: A Model-Based Approach to SCORM Sequencing. In: Proceeding of the Sixth Annual Irish Educational Technology User's Conference (EdTech06) - Research Track, ILTA (2006)
11. Budinsky, F., Brodsky, S.A., Merks, E.: Eclipse Modeling Framework. Pearson Education (2003)
12. Jouault, F., Kurtev, I.: Transforming Models with ATL. In: Proceedings of the Model Transformations in Practice Workshop at MoDELS 2005. (2005)
13. Altova: Altova Xmlspy 2005 User & Reference Manual. Vervante (2005)
14. OMG: Meta Object Facility (MOF) 2.0 Query View Transformation (QVT) (2005) OMG Final Adopted Specification.
15. DeBra, P., Santic, T., Brusilovsky, P.: AHA! meets Interbook, and more... In Richards, G., ed.: Proceedings of the AACE ELearn Conference 2003, AACE (2003) 57–64
16. Power, G., H.C., D., Cristea, A., Stewart, C., Ashman, H.: Goal oriented personalisation with scorm. In: Proceedings of the Fifth International Conference on Advanced Learning Technologies (ICALT2005), IEEE Computer Society (2005) 467– 471