Model-Driven Description and Validation of Composite Learning Content

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

Claus Pahl
Dublin City University, School of Computing
Centre for Next Generation Localization (CNGL)
Dublin 9, Ireland
cpahl@computing.dcu.ie

Abstract: Authoring of learning content for courseware systems is a complex activity requiring the combination of a range of design and validation techniques. We introduce the CAVIAr courseware models allowing for learning content description and validation. Model-based representation and analysis of different concerns such as the subject domain, learning context, resources and instructional design used are key contributors to this integrated solution. Personalised learning is particularly difficult to design as dynamic configurations cannot easily be predicted and tested. A tool-supported technique based on CAVIAr can alleviate this complexity through the validation of a set of pedagogical and non-pedagogical requirements. Courseware validation checks intra- and inter-content relationships and the compliance with requirements and educational theories.

Introduction

The authoring of learning content is a major task. The costs in time and effort have resulted in learning objects (LOs) being introduced as self-contained reusable units of content that not only provide cost-effective, but also quality solutions. Although recent advances in this area have been made to address modelling and composition aspects (Cristea et al., 2007; Dagger et al., 2003; Hummel et al., 2004), there is currently no way to check the pedagogy and structure of composite content defined by the course creator.

Courseware defines a course in terms of its learning content, by defining what learning content to deliver to a learner, when it should be delivered and how. Courseware authoring, also known as courseware construction, is a rapidly evolving research area that is concerned with the tools and methodologies, a course creator uses, to define and create courseware. Courses are increasingly composed of Learning Objects (LOs) which are small, reusable instructional units typically a lesson, assessment quiz, or possibly a tutorial (Wiley, 2001). Through reuse the course creator saves time and money, and can use learning resources that have been tried and tested in other courseware. There has been a move towards standard and specification compliance in defining courseware. The formal separation of learning processes from content in courseware standards and specifications, such as SCORM and IMS LD (Hummel et al., 2004), and the annotation of LOs using a standardised LO metadata, such as IEEE LOM, enables automated courseware validation, as metadata descriptions for courseware and its models can be parsed to ensure the LO satisfies some validation criteria. Courseware specifications define courseware in terms of its components. The componentisation of courseware into a collection of annotated Learning Objects (LOs) presents an opportunity to validate courseware based on its compositional structure.

Courseware validation is a design activity that automatically ensures the presence of certain structural and pedagogical characteristics in constructed courseware (Baldoni et al., 2006; Melia and Pahl, 2009). Courseware validation allows the course creator to minimise the pedagogical problems which the learners must deal with when using immature courseware. Using courseware validation allows the course creator to automatically test for specific pedagogical problems, which may not be possible to check otherwise due to, for example, the adaptive nature of some courseware. This reduces the risk for the course creator.

Our contribution is an investigation of Model-Driven Engineering (MDE) techniques (Schmidt, 2006) to support the authoring of learning content, especially the validation of content composition in the form of learning objects - composition of courseware from individual learning content units (Wiley, 2001). Using CAVIAr allows us to see modelling and constraints specification as MDE activities. MDE can achieve integration and interoperability between different model-based efforts made for content authoring. Integrated models of different authoring concerns play an integral role. We discuss a range of concerns, including the role of models in the authoring process, their application in activities and the benefits for instructor and learner.

Model-Driven Learning Content Description in CAVIAr

The CAVIAr Courseware Authoring Validation Information Architecture shall be used in courseware authoring to automatically validate courseware for a variety of structural and pedagogical concerns including inter-conceptual courseware sequencing (pedagogical concerns regarding the sequencing of concepts in courseware) and intra-conceptual courseware sequencing (pedagogical concerns teaching one concept). Other concerns are the appropriateness of the type of learning material used at particular points in courseware, courseware consistency and aspects of the instructional design in use in the courseware.

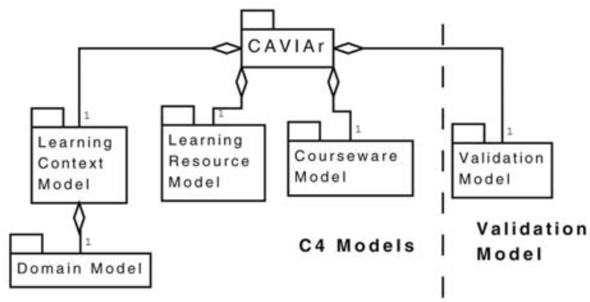

Fig. 1. CAVIAr Courseware Authoring Validation Information Architecture – C4 Models.

C4 model architecture – reflecting content construction concerns

Courseware validation using CAVIAr is achieved by modelling courseware construction concerns. CAVIAr consists of a set of related data models and a validation model. The data models are used to capture the construction concerns used to define and develop courseware (Melia and Pahl, 2008). This set is the CAVIAr Courseware Construction Concern (C4) models, see Fig. 1: the domain model represents the subject domain to be covered by the courseware; the learning context model consists of learner model representations and domain pedagogic information and is responsible for capturing adaptivity concerns that are defined as anticipated learner stereotypes in terms of the domain model; the learning resources model defines the learning resources used in courseware; and the courseware model reflects the courseware structure as constructed by the course creator.

- The Domain Model is represented as a pedagogically neutral conceptual graph. The CAVIAr domain model is used to represent the structure of knowledge that is to be covered in the courseware and beyond. It does this by representing the knowledge as concepts and conceptual relationships (grey part of Fig. 2) where concepts from a database course are primarily related using a taxonomic hierarchy.
- The Learning Context Model, see Fig. 2, defines conceptual sequencing constraints and the learner stereotypes. Each learner stereotype is defined as having assumed initial knowledge (presumed knowledge) and a course goal in terms of domain model concepts, made up of the following information: assumed initial knowledge – knowledge we expect the learner to start the courseware with – and course goals – the knowledge the learner should have after completing the courseware, e.g. ER Modelling, Relational Algebra and Relational Calculus in the example. The model defines knowledge in terms of knowledge type and knowledge level. Knowledge types are defined using Gagné's learning outcomes (Gagné et al., 2005). These are extensions of the domain model, making explicit reference to domain model concepts.
- The Learning Resource Model represents courseware Learning Objects (LOs) and its metadata. Metadata used to describe LOs in CAVIAr is based on the IEEE LOM standard . We would expect here LOs, e.g. covering ER Modelling marked up in LOM format. These form resources in the courseware model below.

- The Courseware Model, see Fig. 3, defines courseware structure and behaviour. The courseware model is defined using courseware topics. Topics contain learning resources to be used by the learner during delivery. Courseware behaviour is defined using conditions that can be placed on topics that define what learners can access that topic and through topic sequencing constraints.

Facetted, i.e. concern-based modelling of the courseware under construction is the starting point, based on which validation takes place.

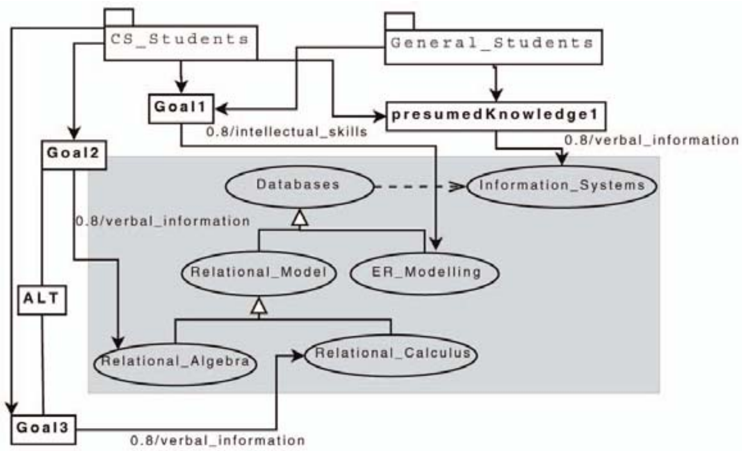

Fig. 2. Sample CAVIAr learning context model with the domain model in grey.

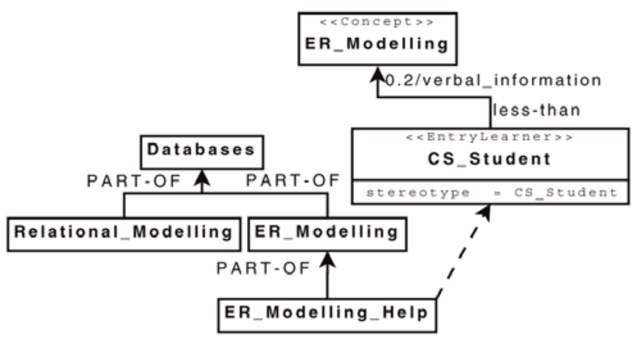

Fig. 3. Sample CAVIAr courseware model depicting a databases courseware.

Model-Driven Learning Content Validation

Courseware validation works by defining constraints in terms of the CAVIAr metamodels – metamodels are definitions that define the structure of the models used above in Figs. 2 and 3, see (Melia and Pahl, 2009) for their formal definition. These constraints must then be adhered to in all instances of the CAVIAr coursware model. The course creator can therefore define constraints that must be true for a courseware model in terms of the CAVIAr metamodel definitions. In determining the CAVIAr validation model, we can split the types of validation constraints into three key categories (see Fig. 4):

- Validation prerequisites. This type of validation checks that data needed for validation are available in the CAVIAr C4 models. The validation prerequisites allows the course creator greater confidence in validation. In the interest of space, we will not look at this category of validation in detail.
- CAVIAr courseware model validation. Validation based solely on the courseware model.
- CAVIAr learning context validation. The learning context model defines the adaptivity and courseware requirements. Using this type of validation, we can check these requirements.

We add a fourth category – validation against an instructional design theory. The courseware model can be validated against formalised principles of an instructional theory.

The validation model is a constraints model which defines valid courseware. The validation model is defined using the Object Constraints Language (OCL), which is an extension of the modelling notation UML we have used so far. OCL is used to define invalid Courseware Model and Learning Resource Model definitions. This can be done using the Domain Model and Learning Context Model. This allows ensuring that conceptual pre-requisite relationships defined in the Learning Context model are adhered to in the Courseware model.

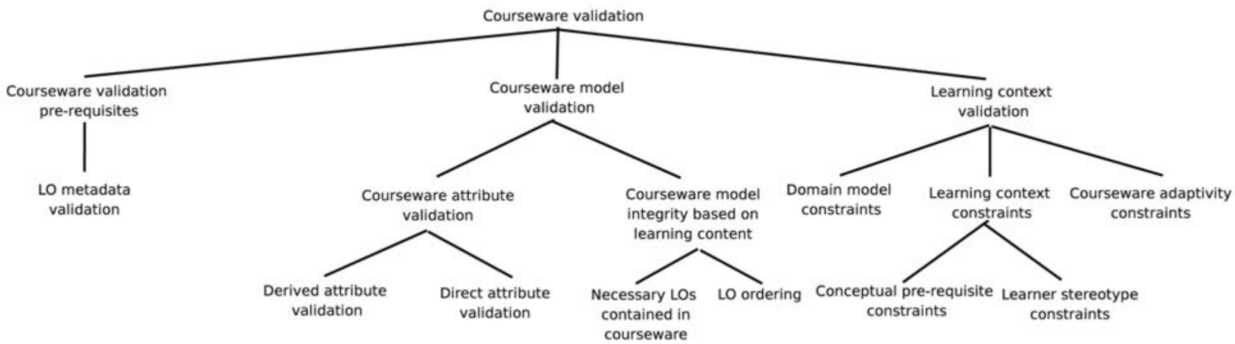

Fig. 4. Classification of CAVIAr validation constraints.

Validation of courseware model integrity. The validation looks to validate the model in isolation from the learning context. Validation based on the courseware model lends itself to two types of validation:

- Courseware attribute validation. This type of validation validates a courseware attribute against an externally defined value. For instance, a duration attribute of the LO could be checked to validate whether an upper limit (e.g. 30 min) or exceeded or not.
- Courseware model integrity based on courseware learning content. This validates the courseware model, ensuring it is structured correctly for the learning content it contains. It for instance checks, if a LO that is reference in the model, actually exists.

Validation of courseware requirements satisfaction. While the first validation perspective was model-internal, we now focus on the satisfaction of external requirements. In validating courseware using the CAVIAr learning context model, our aim is to ensure that the courseware covers the courseware requirements stated in the learning context model. We define three types of instructional constraints using the learning context model:

- instructional constraints using the domain model only, for instance domain model constraints saying that one concept depends on another (e.g. Relational Algebra depends on Relational Model being introduced) need to be reflected in the courseware model.
- instructional constraints using the learning context including the domain model and the learner stereotype information in CAVIAr, for instance prerequisites are validated ensuring that a prerequisite of a specific LO is covered by the LO preceding it (e.g. Relational Algebra is covered before Relational Calculus).
- courseware adaptivity constraints check that entry learner constraints on topics define the correct personalization strategy - entry constraints are defined in terms of learner knowledge and stereotypes.

Validation of instruction design theory. Instructional design theories offer guidance on how people learn and can be applied to ensure that a form of learning occurs, such as constructivism (Reigeluth, 1999). To illustrate how an instructional design theory is validated using CAVIAr, we outline the steps involved in validating that a given courseware uses Reigeluth's Elaboration Theory correctly. Firstly, the elaboration theory is broken down into instructional principles, which must be true for the elaboration theory to be in use. The elaboration theory is defined as instructional principles in (Reigeluth, 1999) as follows (to name two of the principles): 1) tasks are arranged from simple tasks to more complex tasks, starting with the simplest real-world version of the task moving to evermore complex versions of the task; 2) ensure tasks are not too big or too small.
Once the instructional design has been formulated as instructional principles, we can then transform them into instructional constraints in the context of the CAVIAr model. When the course creator has specified the courseware constraints in terms of CAVIAr models, the constraints are then converted to OCL.

```
inv conceptual_prequisites_are_respected:
  self.getAllTopicConcepts()
```

```
    -> iterate(x:Concept;a:Set(Concept)=Set{}|a->union(x.prerequisite(0.5,true)))
  - self.sequencedAfterTopics()
    -> iterate(y:Topic;b:Set(Concept)=Set{}|b->union(y.concepts))
    = Set{}
```

This example illustrates the need to respect prerequisites in topic sequencing. The constraint iterates over topics, making sure that all concepts required by a particular topic (self) are in the set of concepts viewed previously. The validation model can then be validated by an OCL checker, which we have integrated into our environment.

Discussion - Application and Interoperability

Content abstraction and encapsulation through learning objects is a trend that benefits from model-driven construction solutions (using standard modelling notations). Models as abstraction capture essential properties that can be used to construct, but also analyse content composition. Formal reasoning is possible via formal model constraints (Baldoni et al., 2006; Melia and Pahl, 2009). Although requiring some exposure to constraints-based modelling, our user trails have confirmed the relevance of model-driven construction and the importance of validation in the construction process.

Some specific types of courseware systems benefit especially from a model-based validation approach. Adaptive educational hypermedia (AEH) aims to adapt content to the specific needs of individual or groups of learners (Dagger et al., 2003; Cristea et al., 2007). The number and diversity of possible adaptations is difficult to predict and clearly benefits from an automated pre-delivery validation, which we have addressed through the learning context model that covers e.g. stereotypes. Another category of systems are multi-modal courseware systems (Kenny and Pahl, 2005; Pahl et al., 2004) – the sample database course we have referred to falls in this category. Different content types need to be integrated. Although we have not addressed this explicitly here, we can adapt the CAVIAr models to include e.g. interaction mechanisms that need to be reconciled for different media types. This is another concern that can be checked in validation.

A model-driven validation approach can be extended beyond validation in construction. Models can be used to support learning analysis and evaluation at a post-delivery stage, combining models with learner behaviour. These usage mining techniques utilise an explicit model-based structure to interpret mining results regarding the usage of content (Pahl, 2004). Instead of validating the correctness of composed content at a post-construction stage, rich models can actually be used to generate correct content from models without further validation needs (Holohan et al., 2005). This strand builds up on conceptual models and constraints being captured in terms of ontologies. The domain model, for instance, can be enhanced to a domain ontology, if properties and taxonomic and other relationships are added (Boyce et al., 2007). CAVIAr models are extensible due to their explicit meta-models. However, it has to be said that the generation depends on the richness of the models and usually results in content prototypes that need to be further developed by hand.

While, as we tried to clarify, models benefit the construction of learning content, a model-driven approach based on accepted modelling notations and techniques has further advantages. This type of model-driven constructions allows for the interoperability of construction techniques and tools. In (Melia and Pahl, 2008), we have integrated CAVIAr with modelling techniques for AEH. Common representations of central modelling concerns allow us to transform AEH models into CAVIAr and subsequently to validate these models.

The componentisation of learning provides a range of benefits, but also requires challenges to be addressed. Composable, reusable third-party content units – provided as learning objects – form the building blocks of content construction. The validation of these compositions as a quality assurance activity is important. Models as representations of central properties play an integral role in the construction and validation here. Our CAVIAr framework demonstrates that different construction concerns can be captured through different, but integrated model perspectives. Modelling using accepted techniques and notations is a first step towards a more standardised design and construction approach – following on from deployment standards like IEEE LOM.

Acknowledgements

This material is partly based upon works supported by the Science Foundation Ireland under Grant No. 07/CE/I1142 – Centre for Next Generation Localisation CNGL.

References

Baldoni, M., Baroglio, C., Martelli, A., Patti, V., and Torasso, L. (2006). Verifying the compliance of personalized curricula to curricula models in the semantic web. In Proceeding Semantic Web Personalization Workshop at the Third European Semantic Web Conference (ESWC2006). Springer LNCS.

Barrett, R. and Pahl, C. (2004). A Web Services Architecture for Learning Object Discovery and Assembly. International Conference on the World-Wide Web WWW'04. IEEE Press. Pages 446-447.

Boyce, S. and Pahl, C. (2007). The Development of Subject Domain Ontologies for Educational Technology Systems. Journal of Educational Technology and Society (ETS). 10(3):275-288. IEEE.

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

Dagger, D., Conlan, O., and Wade, V. P. (2003). An architecture for candidacy in adaptive elearning systems to faciltate the reuse of learning resources. In Proceedings World Conference on E-Learning E-Learn'03. AACE.

Gagné, R., Wager, W., Golas, K., and Keller, J. (2005). Principles of Instructional Design. Wadsworth, 5th Ed.

Gruhn, V., Pahl, C., and Wever, M. (1995). Data Model Evolution as Basis of Business Process Management. In 14th International Conference on Object-Oriented and Entity Relationship Modelling O-O ER'95, LNCS 1021.

Holohan, E., McMullen, D., Melia, M., and Pahl, C. (2005). Adaptive Courseware Generation based on Semantic Web Technologies. Proceedings International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL2005) at 12th Intl Conf on Artificial Intelligence in Education AIED2005.

Hummel, H., Manderveld, J., Tattersall, C., and Koper, R. (2004). Educational modelling language and learning design: opportunities for instructional reusability and personalised learning. Intl Jrnl Learn Tech, 1(1):110–126.

Kenny, C. and Pahl, C. (2005). Automated tutoring for a database skills training environment. ACM SIGCSE Symposium 2005. St. Louis, US. Pages 58-64. ACM Press.

Melia, M. and Pahl, C. (2009). Constraint-based validation of adaptive e-learning courseware. IEEE Transactions on Learning Technology, 2(1):37–49.

Melia, M. and Pahl, C. (2008). Towards the validation of adaptive educational hypermedia using CAVIAr. In Proceeding of the 6th International Workshop on Authoring Adaptive and Adaptable Hypermedia (A3H2008).

Pahl, C. (2004). Data Mining Technology for the Evaluation of Learning Content Interaction. International Journal on E-Learning (IJEL). 3(4): 47-55. AACE.

Pahl, C., Barrett, R. and Kenny, C. (2004). Supporting Active Database Learning and Training through Interactive Multimedia. International Conference on Innovation and Technology in Computer Science Education ITiCSE'04. ACM.

Reigeluth, C.M. (1999). Instructional Design: Theories and Models, volume 2. Lawrence Erlbaum Associates.

Schmidt, D.C. (2006). Guest editor's introduction: Model-driven engineering. IEEE Computer, 39(2):25–31.

Wiley, D.A. (2001). The Instructional use of Learning Objects, chapter Connecting Learning Objects to Instructional Design Theory: A definition, a metaphor and a taxonomy. Assoc for Ed Comm and Tech.