Utilising Ontology-based Modelling for Learning Content Management

Claus Pahl, Muhammad Javed, Yalemisew M. Abgaz
Centre for Next Generation Localization (CNGL),
School of Computing, Dublin City University, Dublin 9, Ireland
{cpahl|mjaved|yabgaz}@computing.dcu.ie

Abstract: Learning content management needs to support a variety of open, multi-format Web-based software applications. We propose multidimensional, model-based semantic annotation as a way to support the management of access to and change of learning content. We introduce an information architecture model as the central contribution that supports multi-layered learning content structures. We discuss interactive query access, but also change management for multi-layered learning content management. An ontology-enhanced traceability approach is the solution.

Introduction

Learning content management has been a concern that has received increased attention with the emergence of a broad range of content formats supported by Web and Internet technologies. Learning content management systems (LCMS) with different types of differently structured content (e.g. Web pages, animations, executable software applications) require specific solutions dealing with the different representation and structuring mechanisms (Kenny and Pahl, 2005). In addition to this internal view, the external perspective, i.e. access to content, requires support for multi-layered content-centric software systems. Two examples of particular importance recently that can be supported by our key idea of using a traceability approach to link content and content models are:

- access and personalisation – guided access for adaptive retrieval of learning content based on interactive queries through ontology traces (Dagger, 2006; Brusilovsky, 2000; Cristea et al., 2007),
- change and evolution – use of traces for change and evolution management (Gray et al., 2006).

The main focal issues addressed in our investigation are several access dimensions and different content layers. Our approach is ontology-based content management, using a trace model between the ontologies as abstract models and learning content artefacts as the key technique. The central technical contributions of this investigation are

- an information modelling process with concrete modelling activities,
- a formal information architecture model with a trace model at the core.

Our approach adds ontological layers on top of a multi-layered content model, which allows us to reason about content artefacts. We add traces as the primary, processable elements in an integrating, connecting layer – resulting in an ontology-based enhanced trace model for multi-layered content management that supports access and change activities. These techniques result in a significant enhancement for Web-based LCMS where interoperable, sharable semantic content management is needed to deal not only with different content formats, but also access structures and formats that might be shared between different actors.

The paper is organised as follows. We start with the modelling aspects for learning content management applications (based on a case study) in the next section, i.e. a multi-layered LCMS architecture, ontology and information architecture, and information architecture for content management – addressing the modelling process activities and data model definition in each subsection. In the following section, we discuss applications of the information architecture model for two content management activities: access (query and retrieval) and change management. We discuss our approach after that, including related work and end with some conclusions.

Information Architecture for Learning Content Management Systems

Content-Centric Learner Management Systems

Learner management systems are software applications that can be accessed to process learning content. We take a content-centric perspective on these learner management systems here. We consider here content as any digital information – whether static or dynamically created, whether executable or not. Structured and unstructured text, however, will still be the central artefact type.

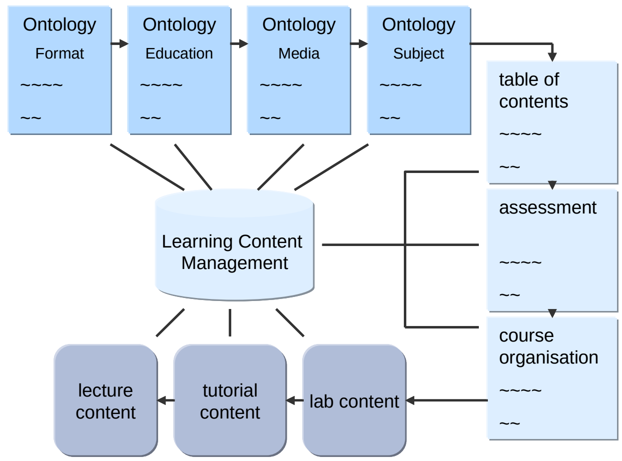

Fig. 1. Layered Content-centric Information Architecture.

We propose a layered information architecture to structure the learning content management systems, see Fig. 1, in which we distinguish between learning content components (bottom), the content management (centre), access mechanisms (right) and ontologies (top) as core artefacts for our case study application. The ontologies support a range of multi-facetted models on the information aspects.

At the centre of our case study is a content management system for an interactive, multimodal learning environment – the Interactive Database Learning Environment IDLE (Pahl et al., 2004). We have in particular investigated the content access layer for the application system. The IDLE system is a database learning environment (subject is databases) for a particular course (format is undergraduate module) that provides content in the form lectures, tutorials and labs (educational formats). Different media types (from text to animations) are used. Based on the ontologies, the learning content management manages content and provides access in different forms (content-centric table of content, assessment elements or organisational information). This system has supported University courses for more than 10 years and has been used by more than 1000 students.

Will illustrate the process of definition such an information architecture model, before looking at access and change management and its support through the information architecture.

Multi-layered Content-centric Application Architecture

The first stage in developing an overall content information architecture is application architecture modelling – central activities here are:

- content artefact identification (content component of a courseware application)
- content-centric system architecture modelling using the content architecture model
- layering of application components (dependencies and references), i.e. relating previously modelled separate application models

These activities are based on the content architecture model (the lowest layer in our architecture, see Fig. 1):

- For artefact identification, we define a set of core artefact types AT = {t1, … , tm} (depending on application domain) and a set of artefacts A = {A1, … , An} with an attribute type(Ai):AT to be defined for each artefact.
- For content architecture modelling, we define a set of components C = {C1, … , Ck} with an attribute partOf(Ci) : A, which are the building blocks of artefacts, and taxonomic relationships H = { isA(Ci1,Ci2), … , isA(Cj1,Cj2) } of subsumption (isA) pairs describing the concept hierarchy.
- For artefact and component layering, artefact relationships types RT = {R1, … , Rl} are defined that define possible relationships (depending on application domain) and concrete artefact relationships R = { Rk(Ai1,Ai2), … , Rl(Aj1,Aj2) }.

The process with its three activities and the model definition shall now be applied to the case study system. *Learning objects* forms the core of the application and consists of the learning system content components, which are accessed through user interface. The application system is IDLE, which consists of the following application-specific components: lecture system, tutorial system and lab system are all learning components (objects). In this learning object context, a second category of components can be identified. Learning environments consist of generic content elements: Web pages (media type used in lectures) as hypertext content, animations (media type used in tutorials) as dynamic content and exercises (used on labs) as interactive content.

The *Learning Content Management* is another subsystem, in addition to the learning objects-based delivery subsystem, which is primarily as a database that organises a support layer on top of the actual delivery application. It is organised based on central access activities such as 'browse table of content' or 'self-assessment' which in turn provide access to content.

Access and Organisational Files, in our case internally structured as XML files, implement the access system by providing different ways to access information. These files provide access to content in order to support some educational activity: the table of content organises content primarily to support the central learning activity; assessment is another central element. The learning activity is about learning the subject, guided by table of contents (a narrative of the content), the assessment activity is about determining the level of comprehension/understanding (reflection), and organisation a an activity comprising of supportive activities to manage learning activities for the learner (self-organisation).

Based on these identifications and categorisations, the content architecture model can be formalised. This structured description of the application system leads to the definition of artefacts A = {learning objects, content management, access files}, which the associated components C as indicated earlier, e.g. type(lecture system) = learning object and part of(Web page) = lecture system. As a taxonomic relationship, isA(animation, tutorial object) can be defined. The artefacts can be related, for instance expressing that implements(Access Files, Content Management) or manages(Content Management, Learning objects).

Ontology and Information Architecture

The next stage of content modelling is ontology-based information architecture modelling (Boyce et al., 2007) – the central activities are:

- ontology identification based on the identified core artefacts (usually the latter serve as the default structure)
- ontology modelling/construction – cross-ontology hierarchies that take in concepts from all central artefacts defining concepts in terms of their taxonomic and other relationships (properties).
- ontology mapping and association – relationships and hierarchy construction – describing the function of each of the ontologies and the corresponding dependencies

Again, a specific data model, the content conceptual ontology model, forms the basis of these activities. This focussed on the upper layer (Fig. 1) – the content architectural model (see above) is included, which through its architectural properties gives guidance for ontology structuring and relating:

- A set of concepts C = { C1, … , Cn } from all artefacts and their components is defined.
- Concept relationships are defined, which include taxonomic relationships H = { (Ci1,Ci2), … , (Cj1,Cj2) } defining a subsumption hierarchy and property types PT = { P1, … , Pm } and properties P = { Pk(Ci1,Ci2), … , Pl(Cj1,Cj2) } describing other generic properties (partOf, etc.) and domain-specific properties.
- Ontology relationships are defined based on concrete artefact relationships R and their relationship types RT.

Similar to the previous section, we analyse the case study application in order to formalise the content conceptual model based on these observations. An *Overall Hierarchy of Concepts* (across different artefacts) captures central concepts and their (mainly taxonomical) hierarchy. We use the inclusion symbol here to denote a hierarchical relationship to relate concepts already identified as components earlier on:

$$\textit{system} \supseteq \textit{learning object} \supseteq \textit{query} \supseteq \textit{topic (types: concept, reference, task)} \supseteq \textit{activity} \supseteq \textit{task}$$

Note that most, but not necessarily all components will be recaptured as concepts. Also, additional concepts that are not content components can be included here to refine the conceptual model.

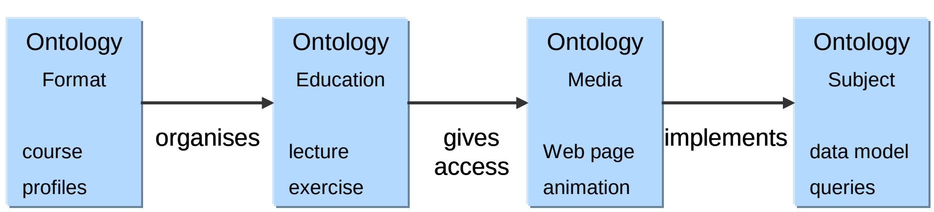

Fig. 2. Ontology Hierarchy.

Ontologies and the *Ontology Hierarchy* emerge from the discussion of the artefacts earlier on. We start bottom-up here (see Fig. 2) with the format, which determines and organises educational activities, which in turn accesses (through the content management) and guides the subject domain-specific learning objects. The content management focuses on learning object components in order to management and locate content, we subject domain components from the function they realise in the educational domain.

The formalisation of the content conceptual model as an ontology is primarily based on the concepts and their organisation into a taxonomy as the core structure. Separate ontologies can be identified (broadly following the artefact default structure here, but merging access files and content management into help infrastructure as they form a coherent domain and splitting software into generic educational and subject-specific parts as two very different vocabularies emerge). For this learning object and content management systems domain, three specific relationships have been introduced (e.g. organises) in order to clarify the roles that individual ontologies play in relation to each other. This will become important later when for instance change impacts have to be determined.

Information Architecture for Content Management

The last stage is the definition of a trace model as the integrating layer between ontological conceptual model (top) and content architectural model (bottom), consisting of:

- trace model definition (based on content artefacts and ontologies) creating an explicit link between the layers, that can become the primary object of manipulation for content access and management activities,
- consideration of ontology-based content annotation in trace generation.

The data model for this layer is a connecting, ontology-enhanced trace model, consisting of:

- content annotation CA = { (A1:t1,l1), … , (An:tn,ln) } as pairs of the type (of content artefact being annotated) and the location with that artefact (or one of its components) – locations depend on the artefact type,
- trace, consisting of source (location), target (ontology) and type (based on type of target element).

Instances in ontology terms are components of the artefacts. Examples from two artefacts are learning object features: a flash sequence is an animation, which in turn is a dynamic content component; or an access file: an entry in a table of contents is a subject domain element, an assignment is an assessment instance. Note that access files as artefacts internally make references (need annotation) to concepts from other ontologies such as the media and domain ontologies. These annotations need to be clearly identified (e.g. through a location ID in for instance text-based content representations) and linked to the content concepts as annotations. The *annotation of content* (access files and content management tables – and also learning objects) through ontology instances links components of artefacts to corresponding concepts. For instance, exercise and multiple choice questions are assessment access components that refer to an educational activity. These annotations need to be made explicit as traces. We can use an example from an access file. These include complex phrases based on basic ontological concepts:

<u>multiple-choice-question</u> <u>assessment</u> on <u>data model</u>

is a learning activity instance that refers to MCQ, assessment, and data model as educational and subject concept instances. These can be represented as annotations CA = {(MCQ:assessment,loc1), (data model:subject, loc2), … } where the locations refer to the word position in the phrase.

Management Activities

The envisioned role of the information model architecture is to support content management activities. Our aim is to illustrate the important role of the trace model as the primary model component. We look at access to content through queries and content retrieval and at the management of change and evolution in these infrastructures.

Access and Personalisation – Query and Adaptive Retrieval

The access systems for the learning objects can be used locally by the learner, but also as a support tool in a learning object repository setting by the instructor. In both cases, user queries describing requested learning objects can be submitted and should result in the assembly of individual components to meet the learning goal expressed in the query. A knowledge-based solution allows the adaptive, contextualised use of information architecture for query processing and information retrieval (Cristea et al, 2007). A query often is of the form "want to learn about models …". A learning object descriptor (educational activity and learning subject) can be inferred. The context of the query is defined by the learning object implementing it. The result determination utilises the ontologies and, if applicable, a user model to support the content management to retrieve suitable learning objects. Adapted learning objects (based on context information) are the resulting output. In this process, firstly, the ontology is used to semantically interpret the query and, secondly, traces are used to identify content artefacts for output assembly.

Change and Evolution

Change will ultimately affect all artefacts – access files being updated, learning objects being improved. The latter will cause knock-on effects on the access files and also the content management (which is the ontology-based trace model). The trace model itself can be used to identify change impacts locally and across artefacts.

We distinguish two categories of changes – changes to the content artefacts (learning objects and content management infrastructure artefacts) and changes to the ontologies as the knowledge on top of the artefact layer (Gruhn et al., 1995). *Ontological Changes* (Plessers et al., 2007) can include adding database language above query language in the taxonomical hierarchy of subject concepts, or Audio could be added below Media Types. Changing query properties in the content management ontologies, for instance tutorials and labs could be 'active learning' as a newly introduced subsuming concept or online submission/correction could be classified as an 'assessment'. Ontological change could reflect changes in the general terminology used. The trace model would indicate further artefacts that are impacted. The trace model acts as a dependency relationship. *Content Changes* can affect any of the artefacts. Of particular interest here are cascading impacts on other artefacts. 1) Learning object changes can have an impact on access files: a new learning object feature (e.g. new media format being used) could result in a new media type (in media ontology) being added, a concept (subject domain ontology), a new format elements (new type of assessment) result in a new educational activity being added. This list is presented in decreasing likelihood of change to the corresponding ontology. 2) Content Management changes could be the addition of a new format type, requiring a new access types in the access ontology, or a new learning object location/identification mechanism, requiring a new ID to be added to the access ontology. 3) Access changes usually do not have structural effects on the ontology, however instance-level changes to the application or software ontology are possible.

Again, the trace model defines the dependency relation that allows determining change impact (Gray et al., 2006). Corresponding algorithms can be formulated (outside the scope of this investigation).

Discussion

The core of our solution is our information architecture – comprising of the modelling process definition and the formalised model. Both the process and the model have been empirically validated by modelling rich educational systems as the one we have used as our case study. The IDLE management (Kenny et al., 2005, Pahl et al., 2004) has caused difficulties and has demonstrated the advantages of modelling. Particularly, adaptive elements of IDLE have benefits (Holohan et al., 2005). Beyond the support in construction and authoring activities, models can have a wider impact. For instance, (Pahl, 2004) discusses the formative evaluation (data mining and usage analysis) as part of a structured change management strategy. Models help to interpret mining results regarding learner behaviour.

Content management has been widely addressed. One specific area is ontology-based change and evolution. Most work has focussed on activities and operators on ontologies (Javed et al., 2009). Qin and Atluri (2009) also make a first step towards content; they investigate the impact of change on the validity of instances. Their work also comprises a formal model, but does not include the extension to semantically annotated content and instances as non-atomic, internally structured artefact. In-depth investigations on learning content management have focussed on adaptive access (Brusilovsky, 2000; Dagger, 2006; Cristea et al., 2007).

With the Future Internet emerging, we can expect a variety of different content artefact type that form Internet applications. We can also expect these different artefacts to be accessed from and shared by different organisations and people. A solution to semantically enhance learning content management that sees these applications as structured into artefact layers can provide significant advancements here.

The main contributions here in this investigation towards this aim are the definition and formalisation of an information architecture model and the description of model development process. An ontology-based solution for the traceability model is the key success factor. The ultimate benefit arises from the use of traceability properties, providing the support for change management activities: targeted, adaptable access having the benefit of accuracy of navigation and retrieval and change management having the benefit of higher automation degrees and predictability.

While the validity of the data models and the adequacy of the process activities have been demonstrated, a number of aspects shall be investigated in the future. We will not only look at different formats, but also the multi-linguality of the Web as a dimension in our content management approach.

Acknowledgement

This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/CE/I1142.

References

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.

Brusilovsky, P. (2000). Concept-based courseware engineering for large scale web-based education. In Davies, G. and Owen, C., editors, Proceedings of WebNet2000, World Conference of the WWW and Internet, 69–74, AACE.

Cristea, A. and Aroyo, L. (2002). Adaptive authoring of adaptive educational hypermedia. In Proceedings of the 2nd International Conference on Adaptive Hypermedia (AH2002), 122–132, Springer-Verlag.

Dagger, D. (2006b). Personalised eLearning Development Environments. PhD thesis, University of Dublin.

Gray, J., Lin, Y., and Zhang, J. (2006). Automating Change Evolution in Model-Driven Engineering. Computer 39(2) 51

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.

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.

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.

Plessers, P., De Troyer, O., Casteleyn, S.: Understanding ontology evolution: A change detection approach. Web Semantics: Science, Services and Agents on the World Wide Web. 5(1) (2007) 39–49

Qin, L., Atluri, V.: Evaluating the validity of data instances against ontology evolution over the semantic web. Information and Software Technology. 51(1) (2009) 83–97

Javed, M., Abgaz, Y., and Pahl, C.. (2009). A Pattern-based Framework of Change Operators for Ontology Evolution. The 4th International Workshop on Ontology Content OnToContent 2009. Springer-Verlag. LNCS.