Adaptive E-Learning Content Generation based on Semantic Web Technology

Edmond Holohan*, Mark Melia, Declan McMullen, Claus Pahl
Dublin City University, School of Computing,
Dublin 9, Ireland.
eholohan@computing.dcu.ie

Abstract. The efficient authoring of learning content is a central problem of courseware engineering. Courseware authors will appreciate the benefits of tools which automate various authoring tasks. We describe a system, *OntAWare,* which provides an environment comprising a set of software tools that support learning content authoring, management and delivery. This system exploits an opportunity provided by the emerging technologies of the Semantic Web movement, most notably knowledge-representation standards and knowledge-processing techniques. The system represents a combination of these newer developments with earlier work in areas such as artificial intelligence (AI) and intelligent tutoring systems (ITS).

A key feature of the authoring environment is the semi-automatic generation of standard e-learning and other courseware elements (learning objects). Widely available standardised knowledge representations (ontologies) and ontology-structured content are used as source material. Standard courseware elements are produced by the application of graph transformations to these ontologies. The resulting products can be hosted by standards-compliant delivery environments.

Adaptivity is an important characteristic of the system as a whole. Authors can select and customise new or existing subject ontologies and employ an appropriate teaching/learning strategy in the generation of learning objects. Instructors can configure the delivery environment either to offer strictly sequenced presentations to students, or to allow also varying degrees of free student navigation, based on the the runtime incorporation of domain ontologies. Students in turn can take the generated courses in the preconfigured delivery environment, and this delivery is dynamically customised to the individual student's preferences and constantly monitored learning track.

The combination of the semi-automatic generation of learning objects with an adaptive delivery environment is a central feature of this new system.
Keywords: courseware generation, Semantic Web, ontology, learning object, adaptivity.

1. Introduction

The key goal in most forms of instruction is the acquisition by the learner of certain *problem-solving* skills. Invariably, this presupposes the lesser goal of *conceptual knowledge* acquisition. Much published instructional training and test materials, including e-learning content, concentrates on this lesser, easier goal [1]. While, in the longer term, we intend to support both categories of learning, in this paper we concentrate mainly on conceptual learning.

The Semantic Web has triggered some new developments in knowledge engineering and machine learning, most notably the standardisation of knowledge specifications. Standardised *ontologies* already serve as shared conceptual-knowledge skeletons, and declarative *knowledge-processing* specifications may soon be used to model tasks requiring common problem-solving skills such as *procedure selection* and *application*. The time is long since ripe for exploiting the above developments for e-learning *courseware engineering* [2, 3]. This project represents some steps in this direction. Specifically, we propose to semi-automate the generation of courseware learning objects by applying graph transformations to the concept graphs represented by the ontologies of the Semantic Web.

The overall structure of this paper reflects the functional architecture of our system. Firstly, we position our work in the broader context of current *of Artificial Intelligence in Education* (AIED) research. Then we describe the basic functionality of our system, *the generation and export of static courseware*. By this we mean the production and export of stand-alone standard learning objects such as slide sequences or tests. Next, we discuss extensions to this functionality in *the generation of courseware for flexible delivery*. By *flexible delivery* here we mean run-time support for combinations of the various forms of *navigation* and *adaptivity* discussed in Section 4. Both Sections 3 and 4 address the production of learning objects for the imparting of conceptual knowledge content. In conclusion, we evaluate our experiences to date and we briefly discuss the possibility of semi-automating also the generation of learning objects that can exercise a learner's problem-solving skills.

2. Background and Related Work

Our work addresses some of the concerns recently expressed by others such as Devedzic [4]. He notes that there are several challenges in improving Web-based education, such as providing for more adaptivity and intelligence, and that a key enabler of this improvement will be the provision of ontological support. The creation of educational Web content with ontological annotation should be supported by ontology-driven authoring tools based on a number of underlying ontologies, e.g., those describing the domain itself, as well as various theories of learning and the instructional design process. Devedzic refers to the the most notable work in the AIED community related to the development of educational ontologies, e.g., that of Mizoguchi and Bourdeau [1], who have outlined a road map towards an ontology-aware authoring system.

Others have taken approaches to concept-based courseware authoring similar to ours, e.g., the *AIMS* system [5, 6] which provides the author with assistance in creating content through domain and instructional models and in configuring these for delivery. A special feature of *AIMS* is the support of a generic set of authoring tasks within the system. A user model captures the learner profile, and provides the information on which adaptivity is based. See also the *Courseware Watchdog* [7] which supports lecturers in the preparation of new courses. It is based on the Edutella peer-to-peer network and employs focused crawling and conceptual browsing to provide personalized access to learning material that is available somewhere on the Web. Another relevant system is *AdaptWeb* [8] which offers adaptive content presentation of Web-based courses, according to selected programs and student's profile. Course contents are customized according to complexity, sequencing, the use of examples and supplementary materials.

3. The Generation and Export of Static Courseware

3.1 The Modularisation of Knowledge and Courseware

Concept graphs, semantic nets or *ontologies* are the natural organising skeletal structure for courseware content. They are similar to popular mind-maps and may be likened to the mental cognitive structures of various pedagogical models [9]. The AI and ITS communities have employed such graph structures for some time and their processing is well understood. But now, thanks to the impetus of the Semantic Web, some globally accepted, interoperable, standard knowledge representations have emerged, e.g., *XML Topic Maps* (XTM), the simpler *Resource Description Framework/Schema* (RDF/S) and the RDF/S-based *Web Ontology Language* (OWL). In addition there is now adequate ontology-processing tool support, e.g., the *Jena RDF Toolkit* from Hewlett-Packard and the *Protégé* GUI ontology editor from Stanford University, together with a widening range of freely available standardised upper and domain ontologies, e.g., the *Standard Upper Ontology* [10].

3.2 The Generation of Learning Objects

The *OntAWare* system contains a tool that accepts as input an RDF/S-based (OWL) ontology file and generates as output a variety of static courseware files, consisting of both interactive and non-interactive learning objects (LOs). The user is required to select one subject ontology from a list of same. For simplicity, let us suppose that the goal of the generated LOs will be to teach the ontology's knowledge content and its class-subclass and class-instance relationships in particular. For illustration, we employ a simple top-down deductive instructional strategy. At the start the author is presented with a menu of target learning concepts (actually the classes in the selected ontology) and s/he can select one of these as the main target learning goal. Then on demand, s/he may automatically generate certain LOs constructed by a standard concept sequencing algorithm, *recursive pre-order depth-first graph traversal* (See [11], [12]), applied to the subtree formed by the selected concept and its related subconcepts. The concept and its description are listed, followed by direct example instances of the concept and their property values. Then, if the concept has direct subconcepts, these are given the same treatment, and so on for their subconcepts too in a recursive fashion.

The non-interactive LOs produced include a lesson plan/outline and a corresponding (e.g., PPT, HTML or other de-facto standard) slideshow sequence, consisting of bullet-pointed slides showing relevant concepts and examples taken directly from the ontology. The above generative graph-traversal algorithm has been modified to produce also an interactive LO in the form of an objective multiple-choice test. For example, this test may include questions such as, *"Which of the following items is (or is not) an example of the concept, X?"* These questions and their answer options are randomly generated, dynamically, and based solely on the interface to the underlying ontology. The objective tests generated by the system conform to the IMS/QTI [13] standards and can be exported to compliant external courseware delivery platforms. The products exported by this component of *OntAWare* would be useful to the student for self-learning or to the courseware author as an aid in the authoring process. Usually, the generated slides would need to be fleshed out further with standard textual or graphic content, and the inter-LO sequencing would need to be determined. The tool provides a demonstration of the technical feasibility of converting standard Semantic-Web ontologies into useful standard learning objects.

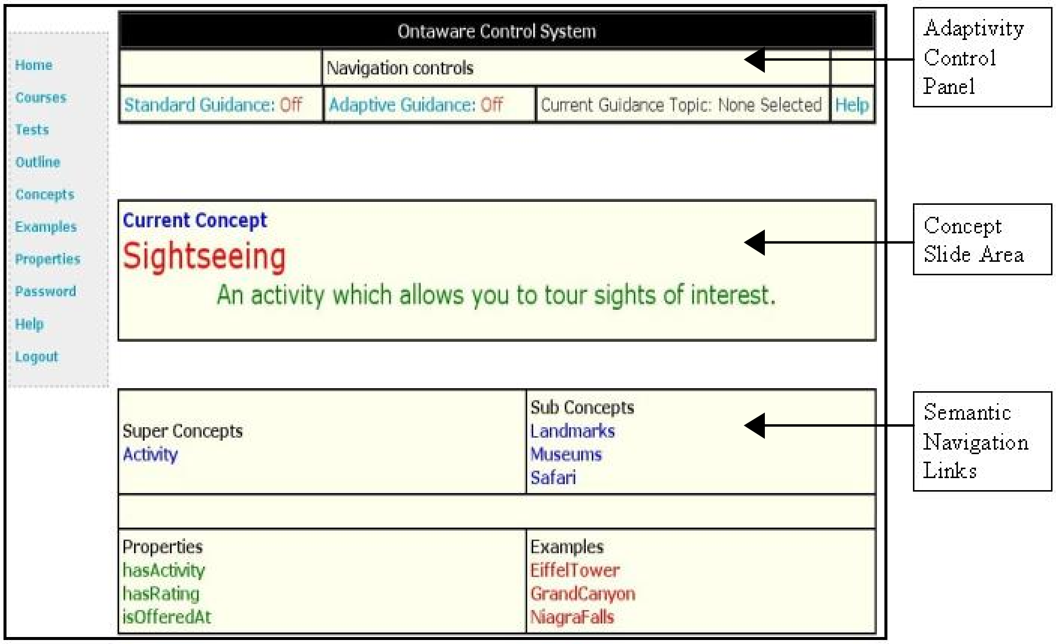

Figure 1: *OntAWare* Active Slide /Concept Navigation Screen

4. The Generation of Courseware for Flexible Delivery

4.1 Constrained, Sequenced E-Learning versus Free Navigation

Semantic-Web ontologies are intended primarily to support free navigation of semantically linked content. In addition to the generation of static courseware, the *OntAWare* system also has its own delivery environment which offers the best of both worlds by supporting totally free ontology-based navigation and allowing the user to specify increasing degrees of constrained navigation based on the generated courseware lesson plans (concept sequences).

4.2 Free Ontology-based Navigation

In addition to the generative functionality described above (3.2), the *OntAWare* system contains a knowledge and courseware delivery environment. In its basic mode of operation this environment offers free ontology-based navigation to the user. At the front end this consists of three types of browser screen layout:

- *Concept Screen* (Figure 1) showing the name and description of the current concept, and links to:
  - Superconcepts of the current concept
  - Subconcepts of the current concept
  - Properties defined for the current concept
  - Individual instances/examples of the current concept
- *Property Screen* showing the name of the current property, and links to:
  - Domain concepts of the current property
  - Range concepts of the current property
- *Individual Screen* showing the name, description and attribute values of the current individual, and links to:

- Concept classes of the current individual
- Property value instances of the current individual

The above screens are dynamically constructed by delivery-tool servlets (Figure 2). These screens, particularly the *concept screen* and the *individual screen*, play a double role in the system. On the one hand they serve as ontology-based content navigation screens, but on the other, they also play the role of *active slides* in the delivery environment. The *slide* metaphor is important educationally, since each slide/screen is intended to focus the user's attention on only one concept at a time and on information directly related to it. The use of these *active slides* requires at least a runtime version of the *OntAWare* system, rather than merely a third-party viewer.

4.3 Option of Guided Navigation

If so desired by the instructor/student, a courseware lesson plan, in the form of guided navigation through the above ontology-based screens, may be activated. In order to generate these lesson plans (i.e., concept sequences), the system currently employs the same ontology graph traversal algorithm described in Subsection 3.2 above. Normally, the system is configured to support two separate categories of user: the author/instructor, who generates the lesson plan, and the student, who follows it. At any stage when following the guidance of a given lesson plan, the student has the option of freely wandering using any of the on-screen links described above. The system constantly monitors the student's movements, so s/he can move forward, backtrack, or return to the generated guided lesson plan at any time.

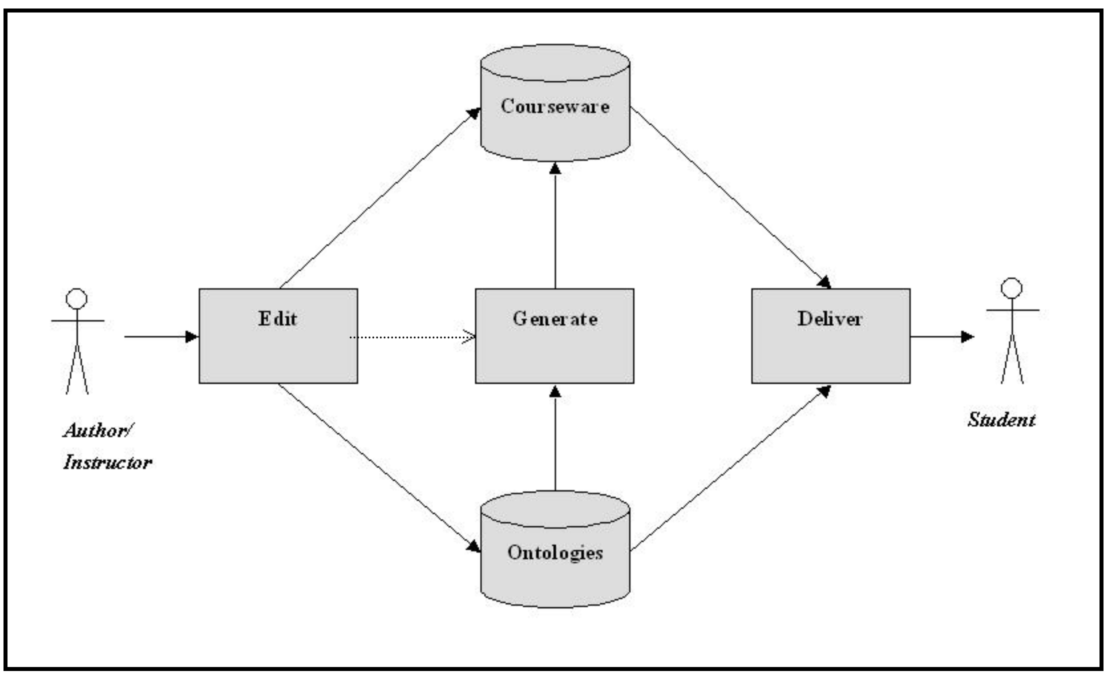

Figure 2: Basic *OntAWare* Functional Architecture.

4.4 Option of Adaptive Guided Navigation

Based on the fact that the student's learning track is now being monitored at lesson delivery time, it is a natural step to introduce some real-time adaptivity into the guidance provided. By default, if *adaptive guidance* is switched on (see Figure 1), the system makes the simplifying assumption, that, if a student has visited a particular concept screen, s/he *knows* that concept and does not need to revisit it. This minimal level of adaptivity can be left under the student's control, or alternatively, the degree of free navigation or of guidance adaptivity allowed to the student may be preset by the instructor. This functionality can be further enhanced by basing adaptivity on student knowledge evaluation using the automatically generated pre- and post-tests mentioned in Subsection 3.2 above.

4.5 Combining Flexibile Delivery with Interoperability

As already noted, by *flexibility* here we mean the options of *free navigation*, *guided navigation* and *adaptive guided navigation* discussed above. Crucially this flexibility is based on the availability of the ontology information at lesson-delivery (i.e., guided-navigation) time. The system also has an advanced feature which allows it to incorporate such flexibility in certain standardised LOs it exports to third-party delivery platforms. We can export whole courses consisting of standardised packages of LOs, e.g., lessons, tests, etc., to third-party delivery platforms. At the same time we preserve the delivery *flexibility* already demonstrated within these LOs. The system uses the SCORM Content Aggregation Model [14] for the packaging of these LOs, together with Simple Sequencing standard for inter-LO sequencing. We do not expose the internals of our generated LOs to third-party environments. Rather, we maintain complete control over intra-LO sequencing/navigation. This implies that a runtime version of *OntAWare*, together with the source ontologies, must be made available within the context of the delivery platform. For greater flexibility learning grid services may be combined with the SCORM runtime environment in the future [15].

4.6 Further Types of Flexibility

At the back end of the system there is support for ontology database maintenance. Courseware learning objects may be generated either from ontologies in the database or from externally sourced ontology files. If so desired, these ontology files can be imported into the *OntAWare* ontology database, and extended by the courseware author with his/her own concepts, examples, etc. Little if any ontology expertise is required of the courseware author, since this is accomplished via user-friendly screens similar to that in Figure 1. This feature provides the basis for further types of flexibility in the system.

For example, this ontology base can act as a master index into a wide range of traditional courseware, web and other content repositories. The courseware author/teacher is allowed to insert his/her own favourite web-page links on the generated screens and these are stored along with the underlying ontology. Tool support to aid link discovery is provided by employing the concept-based metadata searching techniques of the Semantic Web. At lesson presentation time these links may be displayed automatically on the relevant lesson screens. The ability to write to the ontology base also facilitates the implementation of persistent user models, which are overlaid on the underlying ontology, e.g., a simple percentage value can indicate how well the student knows a given concept. Each user's state of domain knowledge can be modelled in this way and updated automatically by the delivery process. This can serve to extend the adaptivity described in 4.4 above.

5. Conclusion

The advent of the *Semantic Web* shows great promise for education and knowledge management generally. However, in addition to the benefits acccruing from the normal uses of this new environment, we believe that the underlying technologies currently being developed to realise the Semantic Web vision can already be exploited in support of courseware engineering.

While others have adopted more theory-driven architectural approaches to the design of similar systems [16, 17], we have consciously taken an incremental practical approach. This strategy has proved to be a fruitful one. First, we verified that we could automatically generate and export useful learning objects from the domain ontologies of the Semantic Web. Second, we developed an ontology-based user-friendly free content navigation system – this then served as the basis for further developments. Third, by combining the first two steps, we introduced ontology-based navigation guidance for the student user of the system. Fourth, we added real-time user monitoring and corresponding adaptive navigation guidance. Fifth, we incorporated convenient transparent ontology modification for the teacher. So far, we have used a simple concept-sequencing algorithm as a representative instructional strategy. However, in the currently ongoing steps we are generalising on this by allowing the teacher to specify in advance the sequencing algorithms and broader instructional strategies to be employed.

The ontology graph transformations employed so far in the *OntAWare* system have been hardcoded (in Java) and so necessarily incorporate an implicit instructional strategy, e.g., the traditional, top-down, deductive, from-abstract-concepts-to-concrete-examples approach. We hope to replace such hard-coded algorithms with declarative ontology-transformation specifications. These will be selected and perhaps customised by the courseware author and used to capture a variety of instructional strategies.

The LOs that can readily be generated from ontology sources necessarily focus on the imparting of knowledge to the student. But the really interesting kinds of LO are those which constructively train the student in some skill requiring knowledge application, e.g., *problem solving* (See: [9], [18], [19]). Future work will examine whether Semantic Web technology can be used to semi-automate the production of such advanced dynamic LOs. Semantic-Web technology consists, not merely of the static source ontologies themselves, but also of the tools used to reason with and transform these ontologies. If we begin by limiting the problem domain to well-defined categories, and describe a given problem scenario stepwise, where each step is defined in terms of a set of preconditions and its successful solution as a set of postconditions, then we may be able to generate LOs that teach or test the skill of solving such a problem. Languages, e.g., OWL-S [20], currently being developed for the standardisation of web service specifications may prove useful here.

So, we expect that ontology graph transformation representations will be doubly useful in the future. Not only will they will provide a method for capturing generalised instructional strategies for knowledge acquisition, but also these graph transformations themselves may well act as sources for the semi-automatic generation of the dynamic learning objects required for skills acquisition.

References

[1] Mizoguchi, R., & Bourdeau, J. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems. In *International Journal of Artificial Intelligence in Education (2000), 11, 107-121.*

[2] Pahl, C., & Holohan, E. (2004). Ontology Technology for the Development and Deployment of Learning Technology Systems – a Survey. *International Conference on Educational Hyper- and Multimedia Edmedia'04. AACE., 2004.*

[3] Holohan, E. (2003). *Automating the Generation of Courseware*, MSc. Dissertation, School of Computing, Dublin City University, 2003.

[4] Devedzic, V. (2004). Education and the Semantic Web. In *International Journal of Artificial Intelligence in Education 14 (2004) 39-65.* IOS Press.

[5] Aroyo, L., Dicheva D. (2004). The New Challenges for E-learning: The Educational Semantic Web. In *Educational Technology & Society, 7(4), pp. 59-69, 2004.*

[6] Aroyo, L., Mizoguchi, R., & Tzolov, C. (2003). OntoAIMS: Ontological Approach to Courseware Authoring. *International Conference on Computers in Education (ICCE'03).*

[7] Staab, S., Studer, R., Stumme, G., Tane, J., & Stumme, G. (2002). Accessing Distributed Learning Repositories through a Courseware Watchdog. *World Conference on E-Learning in Corp., Govt., Health., & Higher Ed. 2002(1), 909-915.* Available: http://dl.aace.org/9460

[8] de Freitas, V., Marçal, V.P., Gasparini, I., Amaral, M.A., Proença Jr., M.L., Brunetto, M.A.C., Pimenta, M.S., Ribeiro, C.H.F.P., de Lima, J.V., & de Oliveira, J.P.M. (2002). AdaptWeb: an Adaptive Web-based Courseware. *ICTE - International Conference On Information And Communication Technologies In Education, 2002. p. 131-134.*

[9] Wiley, D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects: Online Version*. Retrieved Jan 21, 2005, from the World Wide Web: http://reusability.org/read/chapters/wiley.doc

[10] Niles, I., & Pease, A. (2001). Towards a Standard Upper Ontology. In C. Welty & B. Smith (Eds.), *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*.

[11] Fischer, S. (2001). Course and exercise sequencing using metadata in adaptive hypermedia learning systems. In *Journal on Educational Resources in Computing (JERIC) 1(1), 2001.*

[12] Karampiperis, P., & Sampson, D. (2004). Adaptive Instructional Planning using Ontologies. In *Proc. Of the 4th IEEE International Conference on Advanced Learning Technologies ICALT 2004*.

[13] IMS (2002). *IMS Question & Test Interoperability. ASI Best Practice & Implementation Guide -Final Specification Version 1.2.* IMS Global Learning Consortium, Inc.

[14] SCORM (2004). *Sharable Content Object Reference Model*. http://www.adlnet.org (last visited on 11-Feb-2005).

[15] Neumann, F., & Geys, R. (2004). SCORM and the Learning Grid. *4th International LeGE-WG Workshop - Towards a European Learning Grid Infrastructure: Progressing with a European Learning Grid*. Retrieved Feb 8, 2005, from the World Wide Web: http://ewic.bcs.org/conferences/2004/4thlege/session3/paper2.htm

[16] Inaba, A. and Mizoguchi, R. (2004). Learning Design Palette: An Ontology-aware Authoring System for Learning Design. In *Proceedings of ICCE2004*.

[17] Kasai, T., Yamaguchi, H., Nagano, K., and Mizoguchi, R. (2004). Development of a System that Provides Teachers with Useful Resources from Various Viewpoints Based on Ontology. In *Proceedings of ED-MEDIA2004*.

[18] Tecuci1, G., Boicu1, M., & Marcu1, D. (2003). Toward a Disciple-based Mixed-Initiative Cognitive Assistant. In *Proceedings of the IJCAI-03 Workshop on Mixed-Initiative Intelligent Systems, 2003.*

[19] Gašević, D., Djurić, D., Devedžić, V., Damjanović, V. (2004). Approaching OWL and MDA Through Technological Spaces. In *Proceedings of the 3rd Workshop in Software Model Engineering - WiSME2004 at 7th International Conference on the Unified Modeling Language: Modeling Languages and Applications*.

[20] The OWL Services Coalition. (2005). *OWL-S: Semantic Markup for Web Services*. Retrieved Feb 8, 2005, from the World Wide Web: http://www.daml.org/services/owl-s/1.0/owl-s.pdf