Chapter 5
A Case Study of Ontology-Driven Development of Intelligent Educational Systems

Gordon Deline
CBC/Radio-Canada, Canada

Fuhua Lin
Athabasca University, Canada

Dunwei Wen
Athabasca University, Canada

Dragan Gašević
Athabasca University, Canada

Kinshuk
Athabasca University, Canada

ABSTRACT

This article presents a case study of ontology-driven development of intelligent educational systems. Following a review of literature related to ontology development, ontology-driven software development, and traditional software engineering, we developed an ontology-driven software development methodology appropriate for intelligent ontology-driven systems which have ontologies as key execution components, such as e-Advisor, and which is biased toward an integration of incremental and iterative ontology development and downstream Model Driven Architecture for development of software components.

INTRODUCTION

Recent research indicates that ontologies will play a greater role in the development and maintenance of intelligent and adaptive educational systems.

The benefits of ontologies for intelligent educational systems are that (1) intelligent agents can use the developed ontologies as the basis for their knowledge base construction, reasoning and interface design; (2) in a distributed education environment, the ontologies developed for different entities can serve as standardized and open interfaces for interoperability and communication.

DOI: 10.4018/978-1-60960-153-9.ch005

Following review of the e-Advisor system and its development, we looked at how to formalize ontology-driven software construction. Areas of focus included ontology discovery, evaluation, reuse, integration, tie-ins with traditional software engineering methodologies and tools, and integration with governance frameworks.

Using e-Advisor development as a case study, specifically how ontologies are developed and maintained, we have developed an ontology-driven software development methodology that would be appropriate for intelligent ontology-driven systems such as e-Advisor.

This article presents a case study of ontology-driven software development methodology of intelligent educational systems. The organization of the rest of this article is as follows: Section II is a literature review; Section III describes e-Advisor and its ontologies that are key components of the e-Advisor architecture and their development and maintenance; Section IV and Section V review our ontology-driven development methodology as related to e-Advisor; and Section VI relates our conclusions and future work.

LITERATURE REVIEW

As related by Chandrasekaran et. al (1999), ontologies are a key component that can enable effective communication between agents in a multiple agent system, and the building of agents' knowledge. The advantages of ontologies in learner/user models are put forth by Kay (1999) and Chen & Mizoguchi (1999). Razmerita, et al. (2003) put forward an architecture for ontology-based user modeling. Mizoguchi and Bourdeau (2000) discussed the use of ontologies to overcome problems with the use of AI in education.

The W3C's Web Ontology Language (OWL) (Bechhofer et al., 2004) supports the definition of ontologies by building on and adding to the basic support provided by Resource Definition Framework (RDF) and RDF Schema for objects, classes, properties, and hierarchies. OWL supports publishing and sharing of ontologies, knowledge management, and software agents.

Protégé OWL is a tool that is used by domain experts to create and maintain reusable ontologies for knowledge-based systems and is the tool that the e-Advisor ontologies were developed using. As an open platform with a strong user base, a number of useful plug-ins can be found for Protégé to support such functionality as visualization and code generation, as well as the capability to build custom plug-ins. The architecture and features of Protégé-OWL are described by Knublauch (2003); Knublauch, Feargerson, Noy & Musen (2004); and Knublauch, Musen, & Rector (2004). Rector (2003) describes a method for normalization of Description Logics (DL)-based ontologies, specifically those constructed with OWL-DL. Seidenberg & Rector (2006) discuss ontology segmentation techniques and benefits.

Pinto & Martins (2003) give guidelines for qualifying ontologies and enumerate both strict and desirable requirements to aid in this process. Barresi, Rezgui, Lima, & Meziane (2005) outline a methodology for integrating ontologies and Abels, Haak, & Hahn (2005) provide an overview of ontology integration methods categorized as mapping, aligning and merging and Euzenat & Shvaiko (2007) provide more recent coverage of ontology mapping.

Knublauch (2006) makes recommendations applicable for building ontologies with Protégé-OWL and Rector et al. (2004) describe commonly-seen errors and issues with people learning OWL-DL using Protégé OWL and relate best practices and guidelines.

A natural integration point between ontologies and traditional Software Engineering methodologies is the OMG's Model Driven Architecture (MDA) (OMG MDA, 2008), with benefits that can be seen as analogous to the benefits of using ontologies to model domains.

Knublauch (2002) suggested the use of Agile development methodology for the development