Chapter 15
An Ontology-Driven System for E-Learning and Knowledge Management

Gilbert Paquette
LICEF Research Center, Canada

ABSTRACT

• Principles for an Operation System
• Building the Architecture of TELOS
  ◦ Development Process
  ◦ Use Cases and Requirements
  ◦ Conceptual Architecture
• The TELOS Technical Ontology
  ◦ From Conceptual Framework to Conceptual Ontology
  ◦ From Conceptual Ontology to Technical Ontology
• TELOS Main Tools
  ◦ The Resource Manager
  ◦ The Scenario Editor
• Ontology-Driven Scenario Execution
  ◦ The Task Manager
  ◦ Contextual Views
  ◦ Conditions and Control at Run Time

Between 2003 and 2008, within the LORNET research network (www.lornet.org), our team has been designing and developing TELOS, an innovative operation system for eLearning and knowledge management environments that is driven by a technical ontology. After presenting the underlying principles of this system, we will develop a graphic model of the resulting ontology that captures the conceptual architecture of the system. Next, we will present the main aggregation modeling tool and the way it is related to the TELOS Ontology. Finally, we will illustrate how the ontology is used to drive the system at run-time. The conclusion will discuss the contri-

DOI: 10.4018/978-1-61520-839-5.ch015
This has led us naturally to a model-driven, ontology-driven architecture (Kleppe, Warmer and Bast 2003). The main gain of model-driven architectures (MDA) is the generation of the code from the model in successive layers, the model being reusable in other contexts with few adaptions. Ontology-driven architectures (Tetlow et al. 2001; Davies, van Harmelen and Fensel 2002) add to this paradigm an explicit ontology structuring of the objects processed by the system, acting as its executable blueprint. MDA therefore put more emphasis on the platform independent model (PIM), reducing the work on platform specific (PSM) and code models. Ontology-Driven Architectures foster a programming style analogous to the Prolog programming language. Here the declarative part is encoded in the ontology, in our case through OWL-DL statements. The execution part is encoded in queries prepared for an inference engine that processes the queries. The result of a query is to trigger the execution of some of the services.

Another key architectural idea is the concept of multi-actor learning designs and workflows, as the main structure of the various environments produced using TELOS. We wanted to avoid some of the weaknesses of our previous virtual campus models and most commercial platforms, where actors only interact within mono-actor environments that do not really take in account collaborative processes. As we have discussed in Chapter 8, this question is now solved partly in workflows modeling languages such as BPMN (Correal and Marino, 2007) and in eLearning design specifications like IMS-LD (2003) Multi-actor learning designs and workflows provide a central aggregation mechanism grouping actors, the operation they perform and the resources they use or produce from or for other actors. Based on this work, a multi-actor scenario editor and execution engine was planned as a central piece

Recommend this product to your librarian:
www.igi-global.com/e-resources/library-recommendation/?id=1

Related Content

Enhancing Social Networks with Agent and Semantic Web Technologies
www.igi-global.com/chapter/enhancing-social-networks-agent-semantic/65688?camid=4v1a

Semantic Approach to Knowledge Representation and Processing
www.igi-global.com/chapter/semantic-approach-knowledge-representation-processing/35720?camid=4v1a

Web 2.0: Self-Managing System Based on SOA Model and Grid Computing Overlay
www.igi-global.com/chapter/web-self-managing-system-based/39167?camid=4v1a

Entity-Centric Semantic Interoperability
www.igi-global.com/chapter/entity-centric-semantic-interoperability/38036?camid=4v1a