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

bution of this research to the field of ontological engineering of software systems.

15.1 PRINCIPLES FOR AN OPERATIONS' SYSTEM

At the turn of year 2000, new concepts had emerge from various fields such as Web-based programmable learning portals, service oriented frameworks, model-driven and ontology-driven architectures, multi-actor scenarios and workflows. These main technological trends have deeply influenced our work to produce more flexible, powerful, yet user-friendly elearning environments.

We aimed to go one level up, enabling the *aggregation of custom-made platforms* or portals in a way similar to desktop integration that has enabled the interoperation of components from different sources. We have designed the Technology Enhanced Learning Operating System (TELOS) on the same interoperability principles. The TELOS architecture aims to extend portal assembly in ways enabling technologists to built their own platforms. These platforms would foster a variety of distributed learning environments or models such as electronic performance support systems (EPSS), communities of practice, formal on-line training and technology-based classroom, and different forms of blended learning or knowledge management environments.

As the project was starting, *service-oriented frameworks* (Wilson, Blinco and Rehak 2004) such as ELF (2007) or OKI (2007) were proposed to lower the costs of integration, and to encourage more flexibility and simplification of software configurations. Such a framework could also create a broad vocabulary that could be extended to an ontology. The TELOS conceptual framework presented in section 2 would also be designed as a service oriented framework, facilitating the aggregation of services to create custom-made platforms and applications.

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 adaptations. 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