Chapter 2.3
Ontology-Based Software Component Aggregation[1]

Gilbert Paquette
LICEF Research Center, Canada

Anis Masmoudi
LICEF Research Center, Canada

- Software Component Aggregation Process
- Metadata for Software Component Referencing
  - Software Component Metadata (SOCOM)
  - The SOCOM Manager
- The Software Components Ontology
  - The SOCOM MOT+OWL Model
  - The SOCOM Ontology in Protégé
- A Framework for Ontology-Driven Aggregation of Components

The topic of Component-Based Software Development (CBSD) has become very important in industry and research in the last 10 years (Allen & Frost 1998; Object Management Group, 2003). In e-learning, an increasing number of organizations have recognized the importance of building learning technologies by aggregating existent pedagogical software components. To support training processes, Web portals and digitized resources need to be provided to actors in each process. This can be done by aggregating, in a process workflow, different kinds of resources accessible on the Web: documents, simulations, videos, software tools, as well as persons interacting through communication tools and services. These resources are all represented by digital components that need to be aggregated in a proper manner.

Aggregating software components is also a central dimension of the TELOS system that will be presented in chapter 15. In fact, right from the start, the TELOS conceptual architecture documents (Paquette, Rosca, Mihaila & Masmoudi, 2006) proposed a solution to the general resources aggregation problems, whether these resources are actors, documents, learning objects, learning

DOI: 10.4018/978-1-61350-456-7.ch2.3

scenarios or workflows, and of course, software components. These documents suggest a generic framework as a conceptual solution for building e-learning. In this chapter, we present an ontology-based approach for the aggregation of a specific resource type, software components.

Aggregating software components poses many challenges: software component characterization, classification, software component identification, software component integration frameworks, software component integration processes and their assessment (Torchiano, Jaccheri, Srensens and Wang, 2002; Szyperski, Gruntz & Murer, 2002; Bechhofer 2003; Izza 2006).

The effort presented here has been mainly conducted within the Canadian LORNET research initiative on distributed knowledge-based computing for the Semantic Web. LORNET researchers have developed software components and integrated them in a support system, TELOS (Paquette & Magnan, 2008), which support actors involved in learning or knowledge management activities, through the use of resource repository networks on the Semantic Web.

In the first section, we present a short review of related work to justify the presentation of a Software Component Aggregation Process (SOCAP) driven by metadata and ontology models. Metadata management is supported by the SOCOM Web Manager that will be presented in the second section. The metadata presented in this section will help define the SOCOM ontology presented in section 3. Based on this ontology, a software framework will be defined in the concluding section 4, in order to assist software engineer in the component aggregation process presented in section 1.

1 SOFTWARE COMPONENT AGGREGATION PROCESS

Component-Based Software Development (CBSD) is concerned with building complex software systems by integrating previously existing parts called software components. CBSD aims at enhancing the flexibility and maintainability of these systems. It is an approach used to reduce software development costs, and reduce the maintenance burden related to updating large systems (Haines, Carney & James, 1997). The foundation of this approach is the hypothesis that some parts of these systems can be written once rather than many times, and that some software systems can be assembled from existing components, so there is no need to develop them over and over (Allen & Frosts, 1998).

One of the latest trends in systems development is to make greater use of commercial-off-the-shelf (COTS) products. COTS products are commercial components that are ready to use. Component-based systems encompass both COTS products and components acquired through other means, such as non-developmental items. This kind of development becomes feasible due to:

- the increase in the quality and variety of COTS products;
- economic pressures to reduce system development and maintenance costs;
- the evolution and emergence of component integration technologies;
- the increasing amount of existing software that has been designed and implemented to be reused in new development contexts.

Szyperski states that development emphasis moves from programming software to composing software systems (Szyperski, Gruntz, & Murer, 2002). In CBSD. Building systems shifts from writing code from scratch to assembling and integrating existing software components. In contrast to traditional development, component integration is the centrepiece of the CBSD approach. Thus, integration and aggregation are key considerations in the decision to acquire, reuse or build software systems.