Chapter 20

Optimization of WS–BPEL Workflows through Business Process Re-Engineering Patterns

Jonas Buys
University of Antwerp, Belgium

Vincenzo De Florio
University of Antwerp, Belgium

Chris Blondia
University of Antwerp, Belgium

ABSTRACT

With the advent of XML-based SOA, WS-BPEL swiftly became a widely accepted standard for modeling business processes. Although SOA is said to embrace the principle of business agility, BPEL process definitions are still manually crafted into their final executable version. While SOA has proven to be a giant leap forward in building flexible IT systems, this static BPEL workflow model should be enhanced to better sustain continual process evolution. In this paper, the authors discuss the potential for adding business intelligence with respect to business process re-engineering patterns to the system to allow for automatic business process optimization. Furthermore, the paper examines how these re-engineering patterns may be implemented, leveraging techniques that were applied successfully in computer science. Several practical examples illustrate the benefit of such adaptive process models. These preliminary findings indicate that techniques like the re-sequencing and parallelization of instructions, further optimized by introspection, as well as techniques for achieving software fault tolerance, are particularly valuable for optimizing business processes. Finally, the authors elaborate on the design of people-oriented business processes using common human-centric re-engineering patterns.

DOI: 10.4018/978-1-4666-0255-7.ch020
1. INTRODUCTION

A cutthroat competition is currently raging between enterprises in which companies are compelled to constantly evolve in order to realize a competitive advantage. This goal of attaining market leadership is pursued by iteratively altering business processes and strategies aimed at improving operational efficiency (Reldin & Sundling, 2007). Business processes are thus continuously refined, mainly to resolve recurrent issues and as such rectify process performance. This concept is commonly referred to as business process re-engineering (BPR).

Large enterprises have extensively deployed information technology (IT) systems, and have recently started to automate their business processes. Regrettably enough, most of these volatile business processes are enlaced into rigid IT systems and this imposes limitations with respect to the speed with which changes are possible. In the beginning of this decade, this issue led to the concept of service-oriented architectures (SOA) in which IT is flexibly structured to better alleviate the re-engineering of processes by splitting up so-called business logic into a number of software components that are exposed as services (Erl, 2005). With service (operations) as an implementation for individual process activities, a business process can be automated by appropriately orchestrating and coordinating a set of services. Actually this service-oriented computing paradigm has adopted the best practices in distributed computing of - roughly estimated - the past twenty years, and commercially backed by major industry concerns, SOA continues to gain adherence (Stal, 2006).

As one possible SOA implementation technology, web services have managed to become the de facto standard for enterprise software in which various distributed, heterogeneous software systems are integrated in support of corporate e-business and e-commerce activities (Erl, 2005). A web service is typically exposed through a well-defined open XML interface described in the Web Services Description Language (WSDL) document that formally describes the syntax of application-specific messages in XSD Schema format (Microsoft & IBM, 2001) (Erl, 2009). Clients communicate with a web service through an endpoint reference that represents the address and context path where the service is deployed (Microsoft, Sun Microsystems & Computer Associates, 2006). The Web Services Business Process Execution Language (WS-BPEL) XML language is one of the standards that resulted from intensive standardization initiatives by industrial consortia, and shortly became a widely accepted standard for workflow modeling (Abode et al., 2007). The benefit of the central BPEL orchestration component is that the process definition is no longer interwoven inside the implementation code of the business logic. Because of this separation, SOA is said to alleviate the transformation and restructuring of business processes using highly reusable services that can easily be re-orchestrated into BPEL workflows (Erl, 2005).

The service-oriented paradigm turned out to be a giant leap forward in the construction of flexible IT systems indeed. XML-based SOA with BPEL further added to business agility, allowing for the quick development of new business processes leveraging service-wrapped legacy IT assets (i.e., business process redesign). But in spite of the popularity of BPEL and its clear separation of process and business logic, there remain some shortcomings (Buys, 2009; Modafferi, Mussi, & Pernici, 2006). One of these issues is that a BPEL process definition is extremely static: it is designed manually using some software tools and is then loaded into the BPEL engine. Since service orchestration and business processes are at the core of SOA, it is imperative to continuously optimize BPEL process definitions to achieve an increase in system performance, besides having economic implications in realizing a competitive advantage required by the actual continual process evolution.