Chapter 7
Autonomic Workflow and Business Process Modelling for Networked Enterprises

Gerardo Canfora, Giancarlo Tretola, and Eugenio Zimeo

Abstract. As markets become more and more competitive and dynamic, companies need to increase control over their business processes to quickly adapt them to the changing conditions of the operational environment. Workflow management technology is a means to automate and control business processes, but they need more sophisticated capabilities to cope with highly dynamic execution contexts.

This chapter proposes a novel approach to adaptive workflow management, based on a programming model and a related runtime system. By combining imperative and declarative programming, a specific workflow management system is able to react to events sourced from the business environment by modifying the structure and behaviour of running workflows.

The chapter discusses related work on workflow adaptation, illustrates the proposed autonomic workflow model, the overall architecture of the related management system, the technical motivations and choices for the implementation, and the impact of this kind of workflows onto business modelling.

7.1 Introduction

Dynamic markets and globalization force companies to define new requirements for coordinating business actions among supply chains. ICT plays a fundamental role to build virtual organisations based on electronic collaborations [10]. A special kind of e-collaboration is Sense and Respond [4]: supply chains are tuned to collect events and information from the business environment to dynamically alter the configuration and behavior of the supply chain itself according to the modified business requirements.

Gerardo Canfora · Giancarlo Tretola · Eugenio Zimeo
Università degli Studi del Sannio
e-mail: {canfora,tretola,zimeo}@unisannio.it
Workflow management systems (WfMSs) are gaining an increasing importance for handling complex business processes, but they often fail in highly dynamic environments, where imperative behaviours could become unsuitable to address the changing conditions of the running context. This is particularly true in service-oriented computing, where services are handled in an open world and are provided by different organisations: they can be modified or replaced; they can disappear, and new services with different features may become available.

Workflow execution across geographically distributed organisations, technology, and assets introduces the need for features that are beyond the capability of current enterprise management tools and impact the design, supervision and management of workflows. As a consequence, the complexity of IT infrastructures management increases, as they are forced to accommodate for heterogeneous components [26]. Additional complexity is generated by the need for: (i) analyzing a great amount of data collected during execution to improve and adapt processes, (ii) quickly reacting to the evolution in the operating environment, (iii) being able to produce goods or deliver services that are really required, at the right time and in the right place.

This level of complexity is too high for having manual adjustments of workflows during their execution. Autonomic computing (AC) [16], instead, could be an effective approach to automate workflow handling at runtime, since autonomic systems should not only be able to take automated actions, but they should do this with the support of an innate ability to sense, and respond to changes, by incorporating self-learning and self-managing capabilities. Therefore, by considering a business workflow as a large-scale program, autonomic computing could be a key enabler for ensuring self-* properties to workflows.

This chapter presents the main features of a novel approach for dynamically handling workflows, that we define Autonomic Workflows, by exploiting autonomic computing techniques. It provides (i) a programming model that aids users to design autonomic workflows, and (ii) a system that is able to run those workflows with the ability of handling execution anomalies with a limited human intervention. The main objective of the system is to enable workflows management during the entire lifecycle, by collecting and organizing the knowledge from the operating environment. This knowledge, represented in the form of Event Condition Action (ECA) rules, is then used for adapting and improving workflows at run-time.

The rest of the chapter is organized as follows. Section 7.2 discusses research work on workflow adaptation by analyzing in particular the adoption of autonomic computing in the context of workflow management. Section 7.3 presents some typical adaptation situations and defines the conceptual architecture of a workflow management system that is able to handle them. Section 7.4 exploits the model described in the previous section to design a coherent and robust architecture for implementing a workflow management system that is able to seamlessly run autonomic workflows. Section 7.5 shows how the proposed workflow management system can be programmed by using a declarative language based on ECA rules. Finally, Section 7.6 describes how the concept of autonomic workflow can be exploited to model