Chapter 8

Autonomic Business-Driven Dynamic Adaptation of Service-Oriented Systems and the WS-Policy4MASC Support for Such Adaptation

Vladimir Tosic
NICTA, The University of New South Wales, Australia and The University of Western Ontario, Canada

ABSTRACT

When a need for dynamic adaptation of an information technology (IT) system arises, often several alternative approaches can be taken. Maximization of technical quality of service (QoS) metrics (e.g., throughput, availability) need not maximize business value metrics (e.g., profit, customer satisfaction). The goal of autonomic business-driven IT system management (BDIM) is to ensure that operation and adaptation of IT systems maximizes business value metrics, with minimal human intervention. The author presents how his WS-Policy4MASC language for specification of management policies for service-oriented systems supports autonomic BDIM. WS-Policy4MASC extends WS-Policy with new types of policy assertions: goal, action, probability, utility, and meta-policy assertions. Its main distinctive characteristics are description of diverse business value metrics and specification of policy conflict resolution strategies for business value maximization according to various business strategies. The author's decision making algorithms use this additional WS-Policy4MASC information to choose the adaptation approach best from the business viewpoint.

DOI: 10.4018/978-1-4666-1767-4.ch008

INTRODUCTION

In the modern world, technical and business changes are frequent and increasingly common. Additionally, the modern globalized economy increasingly supports and often requires various business relationships between diverse companies. These circumstances place important requirements on enterprise information technology (IT) systems: the ability to seamlessly interconnect with IT systems of diverse business partners irrespective of the implementation of these IT systems and the ability to handle various technical and business changes (e.g., temporary computer/network failures and establishment of new business alliances).

Service-oriented computing (SOC) was developed to answer these challenges. In service-oriented systems (Papazoglou & Georgakopulos, 2003), such as Web services and their compositions, parts of internal IT (e.g., software) systems are exposed as implementation-independent services, which are then composed in a loosely-coupled manner, possibly dynamically, i.e., during run-time (instead of during software/system design). However, just implementing and composing service-oriented systems is not enough to fully address diverse technical and business changes that can affect these systems. To discover and address changes, IT systems have to be managed (Sloman, 1995).

Management of IT systems, including service-oriented systems, is the process of their monitoring and control to ensure correct operation, enforce security, discover and fix problems (such as faults or performance degradations), maximize quality of service (QoS), accommodate changes, and achieve maximal business benefits. Monitoring determines the state of the managed system, e.g., by measuring or calculating QoS metrics, determining presence of problems, evaluating satisfaction of requirements and guarantees, accounting of consumed resources, and calculating prices/penalties to be paid. A QoS metric, such as response time or availability, is a measure of how well a system performs its operations. On the other hand, control puts the managed system into the desired state, by performing run-time adaptation of the system to ensure its correct operation, in spite of changes or run-time problems. For example, control of a service-oriented system includes its re-configuration, re-composition of services, and re-negotiation of contracts between the composed services and between the system and other parties. Control actions result in changes of monitored conditions (e.g., QoS metrics), which closes the management loop.

The majority of past IT system management products act as support tools for human system administrators – these products present summaries of monitored information and often automate some simpler control actions, but it is ultimately human responsibility to make more complex decisions about execution of control actions. Since the complexity of modern IT systems is rapidly increasing, human system administrators exhibit difficulties in making optimal decisions. Furthermore, human system administrators are expensive and might not be available at all times. Therefore, minimizing human involvement in IT system management has been a research goal for several decades and was made prominent by the vision of autonomic computing (Kephart & Chess, 2003). In autonomic computing, IT systems are self-managing, i.e., they manage (e.g., adapt) themselves using configurable policies, with minimal human intervention.

A prerequisite for performing IT system management activities is existence of a machine processable and precise format for specification of the monitored values/conditions and the control actions (Keller & Ludwig, 2003; Tosic, Pagurek, Patel, Esfandiari & Ma, 2005). Policies (Sloman, 1994) are a frequent approach to IT system management, not limited to autonomic computing. A policy formally specifies a collection of high-level, implementation-independent, operation and management goals and/or rules in a human-readable form. To improve flexibility, maintainability, reusability, and simplicity of specifications, policy