Chapter XIX
Distributed Business Rules within Service-Centric Systems

Florian Rosenberg
Technical University Vienna, Austria

Anton Michlmayr
Technical University Vienna, Austria

Christoph Nagl
Technical University Vienna, Austria

Schahram Dustdar
Technical University Vienna, Austria

ABSTRACT

Business rules enable a clear separation of concerns between the core business knowledge and the underlying application code. Service-oriented Computing, on the other side, enables flexible software systems and provides support for business processes based on software services with well-defined interface, descriptions and communication protocols. Yet, the alignment of software services and business rules has not been addressed in literature. In this chapter we present the ViDRE system that bridges the gap between these two paradigms by exposing business rules as Web services. In contrast to existing rule engines, our approach supports distributed rule execution using meta-rules which includes automatic transformation of rules on both client- and server-side.

INTRODUCTION

The design and development of adaptive business applications is a challenging task based on the fact that software applications are increasingly distributed and the requirements change frequently. Such changes involve a multitude of aspects, ranging from changes in the core business logic to changes in the external interfaces to other subsystems or even business partners. Such changes always imply costs and risks that come along while adapting a software system. It is a common goal to keep these costs minimal, therefore, two rapidly emerging trends receive a lot of attention in enterprise computing, namely *service-oriented architecture* (SOA) – or service-oriented computing (SOC) as a research area in general (Papazoglou, Traverso, Dustdar, & Leymann, 2007) – and *business rules*. SOA is an architectural paradigm that enables more flexible software systems and business processes by exposing some functionality as a dedicated software service. Such a service is described with a well-defined, document-centric interface based on well-known interface descriptions as for example the Web services technology using WSDL and SOAP (Weerawarana, Curbera, Leymann, Storey, & Ferguson, 2005). These services are typically distributed and can then be used to orchestrate different services in a business process or to deliver higher-level functionality in composed services (Dustdar & Schreiner, 2005).

Despite all advantages of the SOC paradigm, it does not elaborate on the internals of a service, i.e., how a service is implemented, its technology or paradigms for doing it. In this regard, business rules provide an elegant means to express complex business logic in terms of rules (for example "if-then-else" style production rules) and facts that can be effectively executed by a rule engine. A rule engine is a software component that is capable of executing such rules that are expressed in a given syntax. In the last years, various rule languages and engines have emerged, such as RuleML (The Rule Markup Initiative, 2001), JBoss Drools (JBoss, 2008), ILOG Rules (ILOG, 2008) or Prova (Kozlenkov, Paschke, & Schroeder, 2008). A main drawback of current rule engine implementations is the lack of alignment with service-oriented computing principles, that would allow an easier integration in existing applications by exposing rules as services, so-called *rule services*, and facilitate the creation of *decision services*, that is, a service that is implemented as a rule service with a special semantics of returning either true of false (the outcome of a decision). For example, consider a decision service for calculating insurance claims that accepts a claim as input and decides whether it is approved or denied based on complex calculations and rules.

The ViDRE (Vienna Distributed Rule Engine) project introduced in (Nagl, Rosenberg, & Dustdar, 2006) addresses some existing shortcomings of existing business rule solutions for the use within enterprise software systems that hamper an alignment and integration with the principles of SOAs. One of the main problems is the centralized way in which rules are managed in current business rules management systems (BRMS). Typically, the rules are stored in a central rule base, which poses some challenges when dealing with distributed services that need to access them, especially the increasing network communication overhead. Therefore, we advocate the use of ViDRE *rule service provider*s (VSP), i.e., a rule engine and its set of rules that are needed to execute a dedicated functionality (for example handling an insurance claim). A VSP enables the use of business rules technology in distributed environments by allowing a VSP to use different rule engines that perform the rules execution. This is achieved by a plug-in mechanism that uses the Java Rule Engine API, as developed by the JSR 94 (Java Specification Request) to unify the access to various rule engines (JSR 94, 2004).

Another important issue when using rules technologies is the lack of a standardized rule representation language. When using dedicated