From StPowla processes to SRML models

Laura Bocchi1, Stephen Gorton2 and Stephan Reiff-Marganiec1

1 Department of Computer Science, University of Leicester, Leicester, UK. E-mail: bocchi@mcs.le.ac.uk; srm13@leicester.ac.uk
2 ATX Technologies Limited, London, UK. E-mail: smg24@mcs.le.ac.uk

Abstract. Service Oriented Computing is a paradigm for developing software systems as the composition of a number of services. Services are loosely coupled entities, that can be dynamically published, discovered and invoked over a network. The engineering of such systems presents novel challenges, mostly due to the dynamicity and distributed nature of service-based applications. In this paper, we focus on the modelling of service orchestrations. We discuss the relationship between two languages developed under the SENSORIA project: SRML as a high level modelling language for Service Oriented Architectures, and StPowla as a process-oriented orchestration approach that separates core business processes from system variability at the end-user’s level, where the focus is towards achieving business goals. A fundamental challenge of software engineering is to correctly align business goals with IT strategy, and as such we present an encoding of StPowla to SRML. This provides a formal framework for StPowla and also a separated view of policies representing system variability that is not present in SRML.

Keywords: Service modelling, Policies, workflows, Service oriented architecture.

1. Introduction

Service Oriented Computing (SOC) is a paradigm for developing software systems as the composition of a number of services. Services are loosely coupled entities that can be dynamically published, discovered and invoked over a network. A service is an abstract resource whose invocation triggers a possibly interactive activity (i.e. a session) and that provides some functionality meaningful from the perspective of the business logic [HA04]. A Service Oriented Architecture (SOA) allows services with heterogeneous implementations to interact relying on the same middleware infrastructure. Web Services and the Grid are the most popular implementations of SOA. Exposing software in this way means that applications may outsource some functionalities and be dynamically assembled, leading to massively distributed, interoperable and evolvable systems.

The engineering of service-oriented systems presents novel challenges, mostly due to this dynamicity [WBC07]. In this paper, we focus on the modelling of orchestrations. An orchestration is the description of the executable pattern of service invocations/interactions to follow in order to achieve a business goal.

Considering this view, we encounter both: more typical Computer Science issues as well as a business-oriented perspective—each with their own challenges—but also a need to bridge the gap between the two. In the light of this, we discuss the relationship between two modelling languages for service oriented systems developed in the context of SENSORIA, an IST-FET Integrated Project on Software Engineering for Service-Oriented Overlay Computers: the Sensoria Reference Modelling Language (SRML) [FLB06, ABFL07] and StPowla: the Service-Targeted, Policy-Oriented Workflow Approach [GMRMS07]. The former addresses the Computer Science challenges, whereas the latter focuses on the business-oriented perspective.
Science needs of providing models at a high level of abstraction but with the possibility of presenting quite refined descriptions. The latter is oriented towards the business user. Their combined use provides an approach to bridge the gap between Computer Science and Business needs.

SRML is a high-level modelling language for SOAs whose goal is “to provide a set of primitives that is expressive enough to model applications in the service-oriented paradigm and simple enough to be formalised” [FLB06]. SRML aims at representing the various foundational aspects of SOC (e.g. service composition, dynamic reconfiguration, service level agreement, etc.) within one integrated formal framework. A declarative semantics has been provided in [AF08, LFB07] that maps SRML to mathematical domains that make precise the meaning of the different constructs made available in SRML. In particular, [AF08] provides a formal computational model for SRML which is being mapped into a logic adapted from μUCTL, a formalism being developed within SENSORIA for supporting qualitative analysis [GM05].

StPowla is an approach to process modelling for service-oriented systems. It has three ingredients: workflows to express core processes, services to perform activities and policies to express variability. Workflows are expressed using a graphical notation, such as in [GRM06c] or UML activity diagrams1. Policies can make short-lived changes to a workflow instance, i.e. they last for the duration of the workflow instance and usually will be made during the execution of the instance, rather than applied to the overall workflow model. StPowla allows for both functional and non-functional changes to a workflow, the former have been introduced in [BGR08]. The non-functional changes are called refinement and provide a means to select the most appropriate service based on the current environment when a service is invoked. The functional changes are termed reconfiguration and are short lived structural changes to a workflow instance.

Having considered the two languages in more detail, we can now add a few words to expand on the motivation why both are needed as a way to address the needs of the two communities mentioned above. One can say that SRML is complete in its expressive power with respect to the systems we intend to model. While expressivity is clearly an issue to a Computer Scientist, usability is the more important factor for Business Analysts. StPowla addresses usability partly in making use of graphical notations and more crucially in being modular in that the basic workflow and the policies capturing variability are kept separate while SRML is essentially flat in that it merges both into the same description.

The encoding of StPowla into SRML, on the one hand provides a formal framework to StPowla. Business processes modelled in StPowla can be then represented as SRML models and either being analysed alone or as part of more complex modules, where they are composed with other SRML models with heterogeneous implementations (e.g. SRML models extracted from existing BPEL processes [BHLF07]).

A second reason for the encoding is providing a higher layer to the modelling of orchestrations in SRML that includes a process-based approach to the definition of a workflow schedule, a separated view of policies, that had not been yet considered in SRML, and the inter-relation between workflow and policies.

This paper extends on [BGR08] by introducing a mapping for the refinement aspects of StPowla and providing more detail on the reconfiguration mapping to enhance clarity. Furthermore, we consider an example for the use of SRML and StPowla in combination based on an industrial case study from the SENSORIA project [sen]. Using the example we present the envisaged methodology of using the two languages.

In this paper, we give an overview of the StPowla approach, including the extension for workflow reconfigurations initially proposed in [BGR08] in Sect. 2. We describe the main concepts of SRML, with respect to StPowla in Sect. 3. We then provide an encoding of basic workflow control flow constructs in Sect. 4, and proceed to describe the mapping of StPowla reconfiguration and refinement policies to SRML in Sect. 5. We present the foreseen methodology for the combined use of StPowla and SRML in 6.1 by presenting a case study. Related work and our position relative to these efforts in Sect. 7 and a summary and conclusion in Sect. 8 round the paper off.

2. Specifying and reconfiguring StPowla workflows

In this section, we give a brief introduction to the main concepts of StPowla, highlighting the motivation and then focusing on workflows, and both refinement and reconfiguration policies.

1 http://www.agilemodeling.com/artifacts/activityDiagram.htm