Formal Methods in Model-Driven Development for Service-Oriented and Cloud Computing

Model-Driven Service Monitoring Configuration with Formal SLA Decomposition and Selection

Howard Foster and George Spanoudakis

Department of Computing, School of Informatics,
City University London, Northampton Square,
London, England, United Kingdom
{howard.foster.1,g.e.spanoudakis}@city.ac.uk

Abstract. Configuration of monitoring systems aligned with agreement and guarantees expressed in Service-Level Agreements (SLAs) can be a complex and time-consuming activity. For monitoring functional and non-functional aspects of services, the EU project SLA@SOI has developed a model and language set to describe such SLAs aligning with a service monitoring system architecture for run-time monitoring. In this work we show how these models are formally and mechanically decomposed into terms and primitive events, whilst monitoring component selection for these terms and events is used to generate a complete monitoring system configuration for run-time.

1 Introduction

As a key part of monitoring and management, systems developed with the Service-Oriented Architecture (SOA) design pattern should utilise negotiated agreements between service providers and requesters [1]. Typically the result of this negotiation is specified in Service-Level Agreements (SLAs), which are then used to monitor levels of service provided and to optionally specify pre-conditions and actions to take in the event of such levels being violated. To assist in the steps of service negotiation and monitoring, the EU project SLA@SOI has developed an SLA model and language set [2] which is used to detail SLA Agreement Terms and Guaranteed States (guarantees made by one of the parties involved in the agreement) between service consumer and provider.

In this work we illustrate how these models of SLAs can be formally assessed for *monitorability*, how guarantee expressions can be *decomposed* into terms and primitive events and how monitoring component *selection* can be used to generate monitoring system configurations. This work supports a broader SLA monitoring platform, which offers planning and optimisation, agreement violation detection and overall monitoring system management.

2 Model-Driven Monitoring

Given an SLA model, the process of configuration takes the model (in this case UML in XML format) and translates the SLA Agreement Terms and Guaranteed States in to a set of expressions. The expressions are parsed into Abstract Syntax Trees (ASTs). The ASTs are then traversed for terms and operators which are *matched* with monitoring capabilities that can either provide a computation (reasoners) or provide event information from the services themselves (sensors). Once all Guaranteed States of an Agreement Term have been assessed and matched with features, a complete monitoring configuration can be built. The configuration is then passed to a Low Level Monitoring System for instantiation of the monitors and managing components.

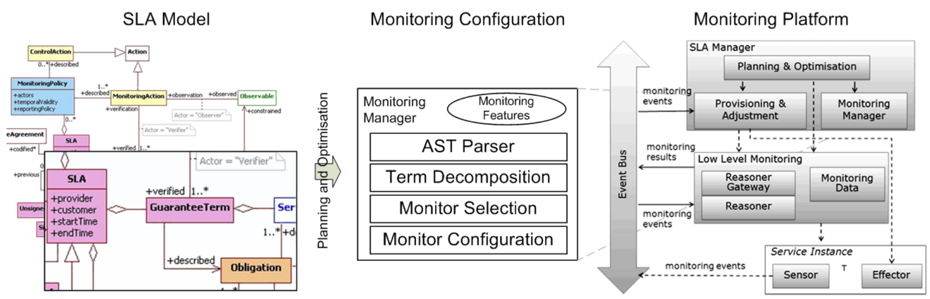

Fig. 1. Model-Driven Approach to SLA Monitoring

3 Decomposition and Selection

The Monitoring Manager abstracts the Guaranteed States (a guarantee made by one of the parties involved in the agreement) that a certain state of affairs will hold for the service. We abstract these states from the Agreement terms and parse the terms using a grammar which is based upon the Backus Normal Form (BNF) specification of the SLA specification. The grammar is used as input to the Java Compiler Compiler (JAVACC) which generates compiler source code to accept and parse source files specified in a defined grammar language. The resulting AST is built to represent the SLA specification terms and expressions. Since the term decomposition is based upon a generated parser, other SLA specification formats may generate their own parsers and transform their SLA specification to the AST input required by the MonitoringManager. In this way, the implementation of the configurator is generic and reusable.

A configuration algorithm is responsible for selecting all the term expressions from the prepared SLA AST tree (TermAST), obtaining a match for the expression terms with available monitoring component features and then building a suitable monitoring system configuration. The algorithm begins by selecting the root of each Agreement Term expression, which in turn holds one or more Guarantee State expressions. An Agreement Term expression is pre-defined as a set of boolean expressions (where all must be true for the Agreement Term to be upheld). Each Guarantee State has a left and right-hand side term and an operator. From the terms a set of input-types is determined. Two term monitors (M1 and M2) are set to reason about the terms and a reasoner monitor set to reason about the expression operator. If the left-hand side of the expression is itself an expression then the second monitor (M2) is recursively configured using the same algorithm (MonitorConfig). If it is not, then the value of the right-hand side of the expression is used as the monitor. Furthermore, a reasoner monitor is assigned to the selection of an appropriate monitor for the input-types, operation and with the list of Features supplied.

4 Configuration

A Monitoring System Configuration (MSC) defines an entire configuration for monitoring an SLA within the monitoring system. The MSC contains a list of components representing Reasoners and Sensors selected to support the Guarantee Terms of the Agreements in the SLA. Each Component in an MSC contains one or more component configurations for each of the different components. For example, an MSC can contain a Reasoner component which has component configurations for two Sensor components and one additional Reasoner component. A Sensor component configuration contains a MonitoringFeature (that which was advertised in selection for the Sensor component) and one or more OutputReceivers. The MSC is generated by traversing the list of selected components discussed in section 3.

5 Conclusions and Future Work

In this work we have illustrated how we leverage a model-driven approach to mechanically configure a services monitoring system from an SLA model. In our approach we accept models of SLAs specifying these expressions, mechanically parse them to formal syntax trees, and formally decompose them to terms and primitive events for effective monitoring component selection. In future work we will explore reallocation of monitoring components for dynamic monitoring configuration and also providing support for optimised component selection.

This research has been supported by the EU Commission under the Framework 7 project SLA@SOI (grant n. 216556).

References

1. Comuzzi, M., Spanoudakis, G.: Dynamic set-up of monitoring infrastructures for service-based systems. In: 25th Annual ACM Symposium on Applied Computing, Track on Service Oriented Architectures and Programming (SAC 2010), Sierre, Switzerland, ACM (2010)
2. Kearney, K. et al.: SLA@SOI Report D5.1a: SLA Model and Template. Available from: http://sla-at-soi.eu/publications/deliverables. (2010)