

Dynamic set-up of Monitoring Infrastructure for SLA Management

George Spanoudakis, School of Informatics, City University London,

January 18, 2010

1. Challenges in monitoring SLAs in Service Based Systems

The paradigm of Service Oriented Computing (SOC) is changing the way IT-based systems are built. Initially, SOC was seen as a way of restructuring the IT stack within an organization in the form of services, and integrating previously non communicating systems through invocations of such services. More recently, however, SOC has evolved into a mechanism for cross-organizational service deployment, in which the systems of an organisation are realised by deploying services offered by other organisations. In this business context, services need to be provided to customers under well-defined conditions. The common approach for specifying such conditions formally is to specify and agree a Service Level Agreement (SLA) between the provider and the customer of a service. An SLA provides a formal specification of the exact conditions (functional and non-functional) under which a service should be delivered to a specific customer (or group of customers) and should be monitored at runtime to ensure that the service provision fulfils it.

Over the last few years, several approaches have been developed to support the monitoring of SLAs. Typically, these approaches collect events during service executions and use them to check whether the properties of service provision as specified in an SLA are satisfied. Such approaches provide state of the art mechanisms for performing the basic checks of service compliance with SLAs but fall short of providing adequate support when replacements of the services deployed in a service based system (SBS) occur at runtime or the terms of the SLA under which a service is provided change dynamically. Such dynamic changes may render the monitoring mechanisms which are used to monitor the terms of an SLA no longer applicable. This can happen for different reasons. A new replacement service, for example, might not be able to provide the runtime events required for monitoring some of the terms in an SLA. Also, after changes in the deployment infrastructure and composition of an existing service, it might no longer be possible to provide the events and monitors for checking the established SLAs for the service. When the deployment of a service is migrated to a new web server which does not support the interception of SOAP messages, it will no longer be possible to execute SLA term checks based on such messages.

To provide effective monitoring support when such changes happen, it is necessary to be able not only to check whether the monitorability of the required SLA terms and conditions is affected by the changes but also to modify the deployed monitoring infrastructure in order to ensure the continuous execution of the required runtime checks. These capabilities, however, are not offered by existing monitoring environments and approaches. To address this gap, SLA@SOI is developing a novel SLA monitoring framework, called "SLA Management for Monitoring" (or briefly SLAM4M). A key characteristic of this framework is the separation of the actual service monitoring from the assessment of SLA monitorability, and the dynamic set up of the monitoring resources (i.e., event captors and monitors) for checking an SLA.

More specifically, SLAM4M groups the activities related to monitorability assessment and the dynamic set up of monitoring infrastructure into a separate monitoring management layer and defines interfaces for integrating this layer with different monitors and event captors. The assessment of the monitorability of a given set of SLA terms by SLAM4M is based on descriptions of the monitoring capabilities of the services that are currently deployed or are going to be deployed in an SBS. Furthermore, to achieve interoperability with different types of monitors and event captors, SLAM4M adopts an event-based monitoring architecture in which monitoring is performed through events captured in the service execution environment by event captors. These events are sent to one or more monitors, which check the satisfaction of SLA terms based on them. In addition to the basic assessment of SLA monitorability, SLAM4M supports the dynamic setup of the service monitoring infrastructure, including the selection of appropriate event captors and monitors, the initiation of communication channels between them, and the delegation of checks of different SLA terms to individual monitors.

2. SLA Management for Monitoring: Overview of the new architecture

A key characteristic of the approach underpinning the design of SLAM4M is the distinction between two key layers in service provision, namely the SLA management and service management layers as illustrated in Figure 1. The SLA management layer is concerned with SLA management activities (e.g. SLA specification, negotiation, modification) and the service management layer is concerned with the software stack required for making a service manageable according to an SLA. From a monitoring perspective, the SLA management layer incorporates the mechanisms required for performing the SLA monitorability checks and the dynamic set up of monitoring infrastructures that can enable the monitoring of an SLA whilst the service management layer incorporates the Event Captors and Monitors required for service event capturing and performing the actual SLA checks, respectively. Given this distinction, SLAM4M belongs to the SLA Management layer, as shown in Figure 1.

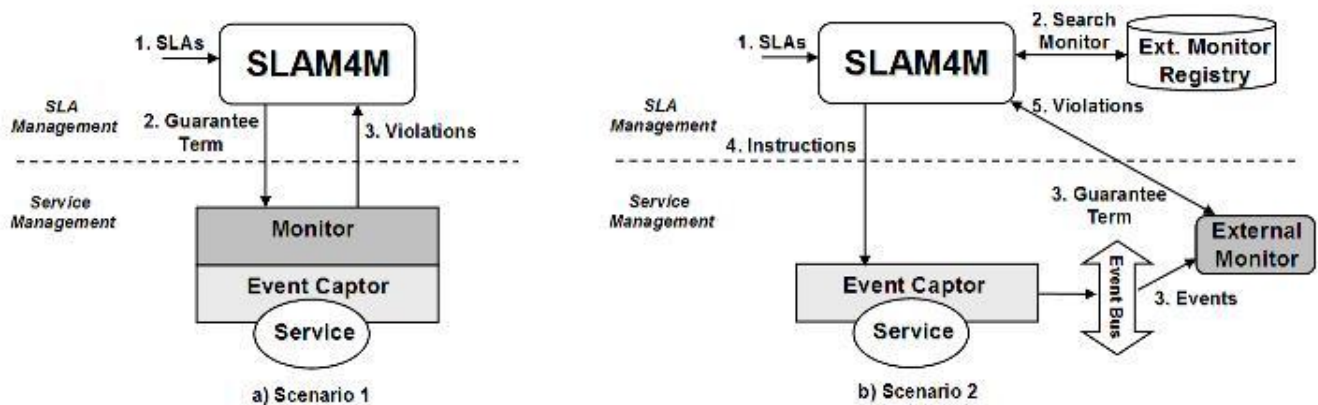


Figure 1 – Scenarios for dynamic setup of monitoring infrastructures

An instance of SLAM4M can, in general, manage one or more atomic or composite services or the composition process (i.e., the “customer” of services) of an SBS. SLAM4M can interact with different event captors and monitors at the service management layer. In general, a service may have different types of event captors that are responsible for capturing and emitting different types of service events. These captors may also have different implementations. An event captor can, for instance, be realized as an instrumentation of the SOAP container or an instrumentation of the BPEL process that realizes a composite service. Similarly, a service may be associated with different monitors which are able to check different properties. Thus, a service may, for instance, have a general purpose monitor that can check functional and non functional service properties (e.g. the generation of specific outputs for given inputs and the average service response time, respectively) as well as specialized monitors that can provide information about the infrastructure in which the service is deployed (e.g., server loads, number of running service instances etc.).

It should also be noted that the guarantee terms of an SLA may need to be monitored by service providers and customers during service execution. Monitoring at the service provider side is important in order to ensure that the provision is according to the SLA and no liability to service customers will arise as a result of deviations from it. At the service customer side, monitoring might also be important to ensure the adherence of the provider to the terms of the agreed SLA or some pre-conditions associated with them related to the service customer. For example, the agreed maximum response time for a service operation in an SLA may be guaranteed only if the number of invocations of the particular operation by the specific customer does not exceed a certain threshold per second. Thus, instances of SLAM4M may exist both at the side of the service customer and the service provider offering monitorability checks and support for the dynamic configuration of monitoring infrastructure to either of these sides.

To support the monitorability checks and the dynamic setup of the service monitoring infrastructures, SLAM4M assumes SLAs expressed in a core SLA modeling model extending WS-Agreement and extracts the guarantee terms of an SLA and matches them with the known monitoring capabilities of a service. These capabilities include the event reporting and the SLA checking capabilities of the service. Event reporting capabilities describe the types of events that can be provided by the event captor(s) associated with the service. Examples of events types required for monitoring include time stamped service operation calls and responses or records of time stamped values of internal process variables for composite services realized by service composition processes. The SLA checking capabilities of a service are provided by the monitor(s) associated with it. These capabilities are represented by the list of SLA guarantee terms specification languages that the monitors of a service support. A monitor is said to support an SLA guarantee term specification language if it can directly monitor terms expressed in this language or it incorporates a mechanism for translating terms expressed in this language into some internal operational monitoring specification. In our prototype, for example, we have used the monitors supporting Event Calculus and RTML rules, respectively.

Hence, at the SLA Management layer, SLAM4M processes SLAs in order to extract their Guarantee Terms, and orchestrates the dynamic setup of the service monitoring infrastructure. To set up a service monitoring infrastructure, SLAM4M retrieves the capabilities of the Event Captor of the managed service and the local and external Monitor engines. On the basis of such capabilities, SLAM4M decides whether an SLA Guarantee Term that is defined for a service can be monitored and, if it can, whether the term will be checked by a local (Scenario 1) or an external service Monitor (Scenario 2). In the latter case, SLAM4M starts the

engagement protocol between the local Event Captor of the service and the External Monitor.

Figure 1 shows the two scenarios for dynamic service monitoring setup in SLAM4M. In the first scenario (see Figure 1a), the managed service is provided with both Event Captor(s) and a local Monitor and has, therefore, both event reporting and SLA checking capabilities. Thus, when it receives an SLA, SLAM4M checks if each guarantee term in it can be monitored locally, according to the capabilities exposed by the Event Captor and the Monitor. In particular, in order for a Guarantee Term to be locally monitored, the Event Captor should be able to provide the required events, while the Monitor should support the language used for expressing the Guarantee Term. The second scenario (see Figure 1b) applies to the following two cases: (i) The Event captor provides the events required for monitoring a Guarantee Term, but the Monitor does not support the Guarantee Term language; and (ii) the managed service has only an Event Captor but no associated local Monitor. In the second scenario, SLAM4M first assesses if the required events are available from the local event captor of the service and then tries to identify an external monitor that can support the Guarantee Term language. This identification takes place through a monitor registry that is accessible to SLAM4M and, if an appropriate external monitor can be found, SLAM4M submits the guarantee term to the external monitor and instructs the event captor of the service to provide events to this monitor. It should be noted that the external monitor may be available at some URI on the network and, therefore, an engagement protocol and an event communication infrastructure are required for establishing and realizing the communication of events between the service event captor and the external monitor.

In the prototype implementation of SLAM4M, we use a “publish/subscribe” event communication infrastructure designated as “Event Bus” in Figure 1b. More specifically, after locating a Monitor, SLAM4M gets from it a token designating an event channel of interest and uses this token to subscribe the monitor to the Event Bus. The same token is passed to the Event Captor to be used when it publishes events to the bus so that these events can be forwarded to the appropriate monitor.

Further details about the monitoring architecture of SLA@SOI, including a discussion of the implementation of SLAM4M and some initial experiences are given in the following article:

Commuzi M., Spanoudakis G. : Dynamic Set Up of Monitoring Infrastructures for Service Based Systems, 25th Annual ACM Symposium on Applied Computing, Track on Service Oriented Architectures and Programming, March 2010 (to appear)