Policy-based Management of QoS in Service Aggregations

Mohan Baruwal Chhetri
Swinburne University of Technology
Center for Complex Software
Systems and Services
Hawthorn, Australia
jmchhetri@swin.edu.au

Bao Quoc Vo
Swinburne University of Technology
Center for Complex Software
Systems and Services
Hawthorn, Australia
bvo@swin.edu.au

Ryszard Kowalczyk
Swinburne University of Technology
Center for Complex Software
Systems and Services
Hawthorn, Australia
rkowalczyk@swin.edu.au

Abstract—We present a policy-centered *QoS meta-model* which can be used by service providers and consumers alike to express capabilities, requirements, constraints, and general management characteristics relevant for SLA establishment in service aggregations. We also provide a *QoS assertion model* which is generic, domain-independent and conforming to the WS-Policy syntax and semantics. Using these two models, assertions over acceptable and required values for QoS properties can be expressed across the different service layers and service roles.

*Keywords*-WS-Policy, QoS meta-model, QoS assertion model

I. INTRODUCTION

Policy-based approaches using the WS-Policy specifications and its extensions have been widely proposed for SLA management including automated service selection [1], SLA negotiation [2], runtime monitoring of service executions [3] and general composite service management [4]. However, not much work has been done on identifying and capturing a complete set of concepts necessary for expressing requirements and capabilities in terms of QoS properties across the different service layers (business, operational/technical service layer and infrastructure/resource layer) and different service roles (such as service aggregator, service consumer and service provider). Additionally, although WS-Policy provides a container structure and simple logic framework allowing policy assertions to be plugged in, a generic assertions model which is rich enough to capture expressive descriptions of QoS requirements and capabilities is missing.

We address the first issue by developing a policy-centered *QoS meta-model* which is rich and expressive enough to capture QoS requirements and capabilities at the different service layers and across the different service roles. We address the second issue by developing a simple, generic *QoS assertion model* which allows the specification of assertions over QoS preferences across the service layers using the QoS meta-model. Assertions can also be made over the relationships and dependencies between QoS properties within and across the service layers and service roles.

Using our policy framework, end-users can express their requirements at any level they are comfortable with i.e. business level, technical service level and resource level. Similarly, service providers can also express their non-functional capabilities and constraints at the appropriate layers. Additionally, the service providers and service aggregators can specify policies which determine how the user-level preferences can be mapped to service level capabilities. This mapping is necessary for both QoS-based service selection and for automated negotiation for SLA establishment.

II. RELEVANCE TO SERVICE AGGREGATION ON THE CLOUD

The Service Oriented Computing (SOC) paradigm has paved the way for a new service-oriented business model which is referred to as *Service Aggregation*. In this model, the service aggregator, which is a business entity, aggregates several component services in innovative ways to generate new business capabilities that it offers to its clients.

This business model is particularly relevant in the cloud computing context where the vision is to offer everything as a service [5], whereby literally any hardware or software resource can be offered as a service. Some example cloud services include *Software as a Service (SAAS)* e.g. Telstra’s TSuite, *Platform as a Service (Paas)* e.g. Microsoft’s Azure and Google’s AppEngine, *Infrastructure as a Service (IaaS)* e.g. Google’s AppEngine, *Storage as a Service (Stass)* e.g. Amazon’s Simple Storage Service, *Data as a Service (Daas)* and potentially *Network as a Service (Naas)*.

Thus a service aggregation on the cloud can include any type of service and the QoS characteristics of these services can be expressed at different layers using different metrics such as business metrics, technical service metrics and resource metrics. The *end-consumers* generally express their QoS requirements using subjective business metrics. The atomic service providers (who are participating in the service aggregation) express their capabilities using objective technical service metrics. The *service aggregators* who are responsible for matching the end-consumer requirements to the composite service metrics and further decomposing them into individual service level metrics have to provide the appropriate mapping from the user level QoS requirements to the technical service level requirements.
III. POLICY MODEL

WS-Policy provides a container structure for clustering or grouping together different alternatives using the WS-Policy operators of wsp:Policy, wsp:All and wsp:ExactlyOne. Each alternative is a conjunction of policy assertions where each assertion represents a requirement or a capability, as shown in the generic policy model in Figure 1, which is largely representative of most policy language proposals. The requirement or capability assertion is a constraint on a given QoS attribute. It specifies the accepted and supported values for the QoS attribute (whether it is a single value, a set of values, an allowable range or a range of values). It uses a predicate operator to describe the constraints.

In order to express requirements and capabilities as WS-Policy assertions, there needs to be an appropriate policy vocabulary which describes the different QoS attributes that are applicable to a given application domain or context. Also the appropriate predicates have to be defined which improve the expressivity of the assertions. The policy-centered QoS meta-model can be used to define the vocabulary for a given application domain/context. The meta-model also supports the specification of preferences over the QoS attributes through the use of concepts such as relative importance and utility.

The assertion model can be used to specify the preferences over QoS requirements and capabilities. In addition, it can be used to specify additional management assertions which aid participants (service consumers and providers) in their decision making process (such as evaluating of offers, ranking offers and generating counter offers) during service selection and negotiation.

IV. POLICY CENTERED QoS META-MODEL

Our QoS meta-model includes all the concepts that are necessary for specifying quality attributes across the different service layers and service roles. Using our QoS meta-model, both service requestors and service providers can specify assertions over their QoS requirements, capabilities and constraints which can be then incorporated into the decision-making process whether it is QoS based selection or negotiation.

Our QoS meta-model provides the following key benefits:

- Most existing QoS models only support simplistic assertions using attribute-value clauses. Our meta-model supports more expressive descriptions through multi-value expressions (such as set of values or ranges)
- WS-Policy currently only supports boolean expressions over hard constraints. Our meta-model can be used to express soft constraints. Since it supports the definition of weighted values (relative importance) and utility for QoS properties it can be used to measure the degree of satisfaction over given offer assertions (as opposed to pure boolean matching only).

V. QoS ASSERTION MODEL

Each policy assertion represents a constraint on the QoS attribute at either the business layer, the operational layer or the resource layer.

A. Preferences Assertion Model

The service requestor can express his/her QoS requirements using the QoS Assertion model. Each QoS Assertion can be for a single QoS attribute or for a group of QoS attributes. Our QoS meta-model is flexible enough to represent user level QoS either as classes of service where each class maps to a specific user-level configuration, or use a more specific approach where the user has the ability to explicitly specify preferences over requirements over each
Figure 3. Preference Assertion over Qualitative Attributes

individual QoS attribute. Figure 3 shows a simple example of a clustering of QoS requirements assertions.

B. Management Assertion Model

The service aggregator knows which services are to be aggregated, how and why. Hence it knows exactly what QoS characteristics are defined for the various services that are aggregated together. Similarly, it knows what user level QoS characteristics to expose to the end users. It also knows how the user level QoS attributes map to the technical service level QoS and resource level QoS. In the management assertion model, we have defined the specific assertions listed below which aid the decision making process. Additional management aspects can be defined as policy assertions as required.

- **QoS Aggregation Assertion** - which defines how to aggregate QoS attributes according to the specific aggregation function, taking into account the service aggregation pattern.

- **QoS Influence Assertion** - which defines the influence relationship a QoS attribute has on others. For example, the throughput has a direct relationship with the price while latency has an inverse relationship.

- **QoS Tendency Assertion** - which defines what the preferred tendency for a given QoS attribute is. For example, for throughput the impact direction is positive and for response time the impact direction is negative.

- **QoS Mapping Assertion** - which defines how the user perceived QoS maps to technical service QoS. Different types of mappings are possible such as one-to-one, one-to-many and many-to-one. For example, the user perceived video quality is qualitative ("Excellent", "Good", "Bad") and maps to the actual media quality which is quantitative and depends upon the technical QoS attributes of frame rate, frame size and resolution as shown in Figure 4.

- **QoS Value Mapping Assertion** - This is an extension to the QoS Mapping Assertion, which allows to map specific values of a QoS attribute at one layer (single values, set of values or ranges) to specific values of other QoS attributes at other layers.

VI. CONCLUSION

In this paper, we have presented a policy-centered QoS meta-model and QoS assertion model for expressing capabilities, requirements and general management characteristics relevant for SLA establishment in a syntax which conforms to WS-Policy specifications. The novel contribution of our QoS meta-model is that it is rich enough to describe QoS characteristics at any service layer and for any service role. Similarly, our generic assertions model can be used to make statements about requirements, capabilities and preferences and constraints over QoS attributes. Our framework aims to complement existing service selection and negotiation frameworks and is particularly useful for enabling automated SLA establishment for service aggregations on the cloud.

REFERENCES


