

Specifying and Constraining Web Services Behaviour through policies

Akhil Sahai, Carol Thomposn, William Vambenepe
Hewlett Packard
{akhil.sahai, carol.thompson, William.vambenepe}@hp.com

Introduction

In the context of web services, policy provides the means for specifying and modulating the behavior of the service to align its capabilities and constraints with the requirements of its consumer. All facets of the behavior of the web service should be modulated by policies that are classified in terms of the manageable aspects of the service, and that achieve alignment between the objectives of the producer and the consumer.

Policy can be thought of as the entire set of strict (enforced) constraints and desirable directives that control the behavior of a target entity (service) towards achieving a goal. Policies can be used to govern virtually any aspect of service behavior. Policy management involves a wide-range of functionality. The set of functionalities that policy management comprises of are:

- § Policy specification: specification of policy in clear unambiguous manner, also involves translation of higher-level goals and objectives to lower level policy specifications, creation of a Policy framework that can be used to deploy, retract policies [2][3], Policy language(s) to specify policies in [4], specification of actual Policy Constraints and Rules, Policy Actions that are related to constraints, flows, Meta-policies and policy transformation
- § Policy conflict-detection and resolution : handling Inter-domain policy constraints, Intra-domain policy constraints, and dealing with Constraint conflicts, Action conflicts
- § Policy translation and deployment : Mapping them onto the architecture and systems, Policy feedback (handling traps, alarms evaluating affected policies and triggering actions, Policy definition points/deployment points
- § Policy violation detection and control: monitoring metrics and value changes, detecting existing or impending policy violations, enforcement of policies that may involve actions.

Policies in the Adaptive Enterprise

HP's vision for the Adaptive Enterprise is to synchronize IT with business to manage change. This synchronization requires that operational policies of IT be in alignment with the business strategy and objectives for the enterprise. An Adaptive Enterprise delivers IT services through multiple layers. Each of these layers has their own models, data sources, metrics and policies specified in multiple formats. For example,

- § Infrastructure Virtualization Layer: The resource level information is modeled in terms of standards such as Common Information Model (CIM), as well as through models specific to a particular domain or function.
- § Web Services Layer: The Web Services layer has its own modeling language: Web Service Description Language, augmented with specifications from OASIS such as WS-ResourceProperties and WS-BaseNotification.
- § Business Process Layer: The business Process layer is modeled in terms of WS-Choreography, BPEL4WS (Business Process Language for Web Services).
- § Business Layer: This layer uses technologies like business score cards to model metrics. The business strategies constitute the highest level of policy, and drive the policy specification through the layers of IT services.

Policies at these layers are specified in a multitude of ways, e.g. CIM, PARLAY, IETF, Ponder, WS-Policy, XACML etc. For each layer the metric sources are different and the instrumentations are varied.

The realization of the vision of adaptive enterprise implies the following considerations for policy management:

- § Policy specification: Typically a large number of technologies and products are involved in resource life-cycle management from design, deployment, monitoring, and re-design to decommissioning of the system. It is important to specify policies in a coherent manner so that one specification can be translated into another
- § Policy conflict-detection and reasoning: Since there are multiple stake holders, namely operators, administrators, users, and business managers, each of whom may have conflicting requirements on the system, it is important that policies can be specified without complete knowledge of the system, and there should be mechanisms for detecting conflicts in policies at Policy Decision Points (PDPs).
- § Policy translation: The translation of policies, metrics and data from one layer to another is done usually in an ad-hoc and haphazard manner. These policies must be translated across the multiple layers of the enterprise so that higher-level business objectives may be translated to infrastructure resource level requirements ,and lower level policies may be aggregated into higher level policies.
- § Policy violation detection and enforcement: Policy Enforcement Points (PEPs) must support policy-related measurements as well as policy enforcement.

Because Web Services play an important role in the adaptive enterprise, it is important to define policies for Web Services in a way that allows them to be synchronized with the business objectives of the enterprise.

Policies in Web Services

Web services hold the promise of bringing greater agility to the enterprise. In order to deliver on that promise, a number of fundamental issues must be addressed ,beyond agreement on document exchange formats. This involves both the B2C and B2B scenarios where users may access web services over the Internet or web services may interact with other web services.

Some of the issues specifically relevant to web service composition, are:

- § How will web services agree on the nature of the interaction (i.e. what will be provided by each of the participants)?
- § How will web services agree upon how well (performance, quality, etc) each of them will execute?
- § Who will be responsible for the overall execution or completion?
- § Who will be responsible if there is a failure in the overall execution or completion?
- § How will web services trust each other?

Policies are the cornerstone for addressing these issues in web services. Policies may be used to guide the interactions between users and web services or between web services. These policies may determine what transaction need to be executed and how well they should be executed. They will also help in allocating responsibilities and risk among web services. In a real-world scenario each web service may interact with many other web services, switching between the roles of being a provider in some interactions and consumer in others. Each of these interactions must be governed by policies .

Emerging standards such as Web Services Definition Language (WSDL), WS-ResourceProperties, WS-BaseNotification and BPEL4WS are creating flexible and generic interaction models for web services. For example, WSDL introduces concepts such as operations, ports, and end points – which are useful for describing the operations of any web service. Similarly, BPEL4WS introduces the notion of activities and flows – which are useful for describing both local business process flows and global flow of messages between multiple web services. WS-ResouceProperties allows acces to properties of resources exposed via Web services and WS-BaseNotification (along with WS-Topics) introduces the concepts of subscriptions and notifications. So, one way to create a flexible policy formalization is to build upon these concepts. In other words, one can create a flexible policy formalization by associating “constructs” to the formalizations that are already defined in WSDL, WS-RP, WS-BaseNotification and BPEL4WS. Here are some examples that show how such association could be applied.

- § Response time of a web service operation.
- § Response time of a flow.
- § Security of an operation.
- § Number of times an activity is executed in a flow.
- § Cost of executing an operation.

- § Availability of an end point.
- § Access rights to a property (read/write)
- § Mutability of a property
- § Maximum number of notifications sent per hour

Policies in web services will usually deal with Security (Authentication and Authorization), Quality of Service, Reliable Messaging, Privacy, or capabilities and constraints specific to a particular service domain. These policies will guide how web services behave and compose with other web services or are used by other clients that use XML/SOAP for interacting with them.. For example, if there are policies specified on two web services that refer to their corresponding operations and type of authentication mechanisms that they expect (X.509, or login/password) the two conversing web services must have compatible policies that guide the type of authentication information the consumer service can send out and the type of authentication information the provider service can support. In addition the policies have to be specified in such a way that these composition processes can be automated and reasoned about[1][5] and conflicts in policies can be determined. Similarly expectations and capabilities to support response times on operations/processes have to be expressed as policies on either of the web services that have to be compatible with each other before the web services can compose with each other.

In summary,. Policy-based management of web services plays an important role that in HP's adaptive enterprise vision. Also Policies are critical for determining web services behaviour . They have to be specified so that they can refer to all the constructs in the corresponding namespaces defined in their WSDL, BPEL4WS specifications and have to be specified in such a way that they are capable of expressing complex constraints, and can be reasoned/analysed /evaluated to automate web service behaviour.

References

- [1] Akhil Sahai, Sharad Singhal, Rajiv Joshi, Vijay Machiraju. Automated Configuration Generation through Policies. IEEE Policy 2004. Also HPL-2004-55
- [2]DMTF Policy
http://www.dmtf.org/standards/documents/CIM/CIM_Schema26/CIM_Policy26.pdf
- [3] PARLAY <http://www.parlay.org>
- [4] Nicodemos Damianou, Narankar Dulay, Emil Lupu, Morris Sloman: The Ponder Policy Specification Language. IEEE POLICY 2001: 18-38
- [5] Object Constraint Language (OCL).
<http://www-3.ibm.com/software/awdtools/library/standards/ocl.html#more>