

Quality-driven Design and Management of Service-oriented Software Systems^{*†}

Tan Phan

Faculty of ICT, Swinburne University of Technology, 3122 Hawthorn, Australia
tphan@ict.swin.edu.au

Supervised by Jun Han, Jean-Guy Schneider (Swinburne University of Technology) and Steven Versteeg (CA Labs Melbourne)

Abstract. Aligning SOA service and system properties with original business requirements during service design and operation is a major challenge that current research has not addressed in full. In this PhD work, we introduce the HOPE (High-level Objective-based Policy for Enterprises) framework that supports in a systematic manner the specification of quality-oriented policies at the business level and their refinement into policies at the system/service level. Our work is also aimed at defining an effective mechanism for business-oriented runtime monitoring of system operations and service interactions for quality conformance. Our further objective is to define adaptation mechanisms to overcome non-compliance. Our focus is on the security domain. Central to our approach is a service registry which acts as a facility for the management of policy lifecycle, to maintain the association of high-level business policies, quality objectives, and system level policies.

1 Background and Motivation

Business rules and regulations from regulatory standards such as SOX 404 [1] control the operation of many business processes and thus constrain the development and usage of IT systems that support those processes, which include Web Service(Ws)-based SOA systems. While many business rules and regulations must be translated into functional requirements for such software systems, others can be translated into quality requirements, such as those concerning security, availability and manageability. These requirements can be formulated as high-level quality objectives, e.g., “*Customer data must be kept confidential*” and realized using various means of IT management and governance.

To ensure that WS-based SOA systems are interoperable and dependable, various industry standards have been proposed to support the specification and management of quality aspects of WS such as those about security, reliable messaging, and transactions. In general, these standards are about system-level

* This work is supported by the Australian Research Council and CA Labs.

† The author would like to thank Ingo Muller for his valuable comments for this paper

mechanisms used to achieve some non-functional qualities. Example mechanisms in security are role-based access control and message encryption and signing. The WS Policy framework (WS-Policy) [2] is a standard that supports the specification of quality properties for Web Services and service systems. Such standards, however, focus on describing low-level technical details for governing service interactions, rather than on specifying high-level, business-oriented requirements.

One of the issues that needs to be addressed is how to *align the high-level, business-oriented quality objectives with the system-level realization mechanisms* offered by WS standards such as WS-Policy. Currently, the high-level quality objectives are often identified by practitioners such as business analysts or IT compliance officers who often do not have an in-depth understanding of all the system-level realization mechanisms for such quality objectives. It is the system developers who are responsible for realising them. This realisation process is rather ad hoc and it is thus difficult to ensure that a system fully possesses all the required properties. As such, a contribution of great value would be a systematic process and related techniques that can *derive* the system-level realization from the business-level requirements and can verify that the realization actually fulfils the requirements.

Once we have come up with a set of design-time system-level realization mechanisms for the quality objectives, a step further is to address the issue of how to guarantee that the realization mechanisms are actually fulfilled when the system is in operation. Being able to select the appropriate quality aspects to monitor and being able to map the monitored events to the original requirements would increase the chance of identifying and resolving non-conformance. An adaptation mechanism which can analyse the quality non-conformance and derive a set of changes to be performed is needed to ensure that the requirements are always respected.

Related Work. While SOA governance is a very active research field, many of the issues identified above have not been addressed in full. Firstly, even though there exist a number of policy frameworks and languages with associated refinement and management techniques such as Ponder [3] and KAoS [4], none of them are readily applicable for specifying SOA. In particular, most of the existing frameworks fall short in enabling the specification of business requirements and the refinement of them into system-level policies. More details can be found in our review paper in [5].

There have been a number of attempts to apply model-driven architecture (MDA) techniques for the modelling and translation of SOA qualities into system-level realization mechanisms such as [6] or [7]. In such work, quality properties of services and applications are modelled in platform-independent manners which are then transformed into platform-dependent codes and configurations for middle-wares to realize these qualities. However the entities being modelled are technical entities, representing technical concepts like *filter*, *connector*, *services*, and *proxies*, not business-oriented entities. This not only limits the participation of business analysts and IT compliance officers in the modelling process but also makes it hard to align the models with the original business requirements.

It is also seen that even though there are different approaches for WS management, the support for business-oriented management is not adequate. Current research work in the field of runtime service management focuses more on the system-level management. Various techniques have been proposed for the specification of service properties and the monitoring and management of them such

as in WSLA [8], SLANG [9], and WSMN [10]. However, little has been done to allow for the management of services from a business perspective even though the importance of business-oriented service management has been acknowledged such as in [11]. The main focus of work in this area has been on specifying and enforcing service SLA. We are unaware of any work that attempts to incorporate other business aspects such as compliance to standards, rules and regulations.

Research objectives. The main objectives and primary contributions of this thesis are as follow 1) We define a general framework for the specification of high-level quality requirements for systems and present a mechanism to refine them to system-level realization mechanisms 2) We provide an approach for policy-based service registration and discovery with algorithms and techniques for detecting non-compliance of services to organizations' quality requirements and for verifying service-client quality requirement compatibility 3) We aim to provide a novel monitoring mechanism that can map service runtime interactions back to the original business requirements in an intuitive manner. For this objective, we consider the use of techniques such as Finite State Automata for modelling the requirements and Bayesian network for failure analysis 4) We aim to define techniques and algorithms for generating actions that can be performed to guarantee compliance to quality requirements. We consider applying techniques in the field of Autonomic computing for this. For all of these, our focus is on the alignment of business requirements and SOA systems and we use security as the example domain for our approach.

2 The Approach and the HOPE Framework

The approach. We propose the High-level Objective-based Policy for Enterprises (HOPE) framework which is aimed at addressing the above research problems via policy-based design and management. As presented in Figure 1, with HOPE, we specify quality requirements (which are driven by original business rules and regulations) in the form of business-level policies which will then be formulated as quality objectives applicable on business entities. The objectives are then refined into *system-level* Web Services policy for Web Services-based applications. Such policies are used to regulate runtime service interactions and provide information for adaptation in case of policy non-compliance.

Central to the approach is a *service registry* which acts as a facility for the management of policy lifecycle including the *modelling, analysis and design, creation, usage, update, removal* of policies; and maintain the association of high-level business policies, quality objectives, and system-level policies. The *registry* provides a point of reference for various design time and runtime management operators such as *WS-Monitor and WS-Enforcer* to retrieve policy-related information and store the relevant data that they collect.

Our approach employs policy-based management which has the advantages of being able to dynamically update the behaviour of a managed system according to the changing context requirements without having to modify the implementation of the managed system. Also, the declarative specification of rules and regulations in the form of policy statements are more concise, intuitive and simpler to verify than

procedural code. Furthermore, service registries hold service metadata and are characterized by rich metadata management and rich query capabilities. As policy is one important type of SOA metadata, service registries' capabilities can be extended for policy-based management.

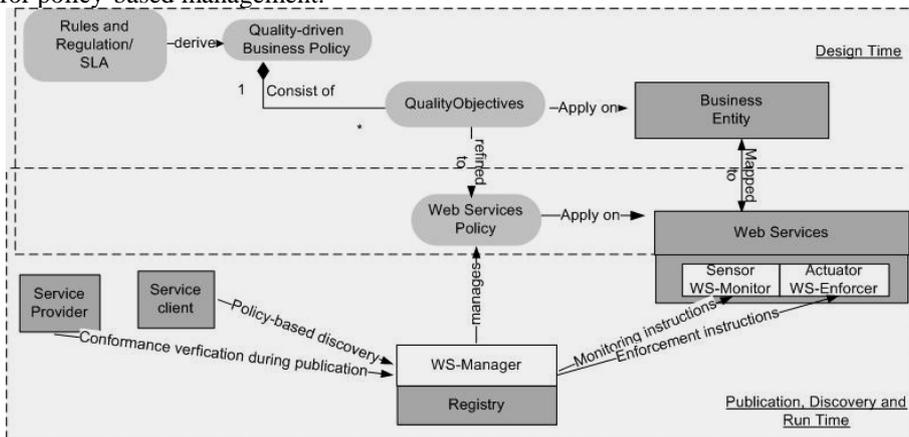


Fig. 1. Policy-based SOA quality management using a service registry

Validation of the approach. To validate the work we use a business case study together with research prototyping. We design a business case scenario and examine a database of global rules and regulations from the Unified Compliance Framework (<http://www.unifiedcompliance.com/>) to identify applicable business requirements. We then, from such rules and regulations, using our approach to derive business policies and use such policies to validate our refinement, monitoring and adaptation techniques. A prototype for HOPE is also being built and once the tool is ready, we plan to present it to a group of developers and business analysts for validation.

Work to date

During the past 18 months, I have worked on addressing a number of research issues, aligning with research objectives (1) and (2) presented above. Details are as follows.

Quality-oriented business policy specification and refinement. We investigated the research issue of quality-driven business policy specification and refinement of SOA Systems in [12]. In this work, we proposed a framework (Figure 2) that supports the specification of business level quality-oriented policies and their refinement into policies at the system/service level.

As can be seen in Figure 2, in our approach, quality-oriented business requirements (quality requirements) are expressed as quality objectives applied to business entities which are modelled in application entity model. These objectives are then refined or translated into *system-level* WS-Policy statements. The refinement relies on an application-specific business entity model and application-independent domain quality models, for which we created the meta-models. We illustrated the approach with a *Mortgage loan approval* business case study to demonstrate the policy specification and refinement for qualities in the security domain and have implemented a proof of concept prototype.

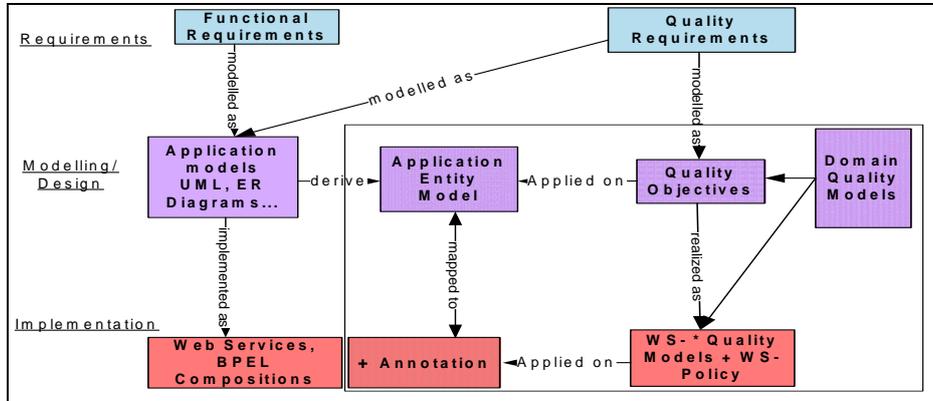


Fig. 2. A framework for quality-driven policy specification and refinement

Policy-based service registration and discovery. Another research issue that we have identified and addressed was the issue of policy-based registration and discovery [13]. In [13], we argued that (1) ensuring service qualities (specified in the form of WS policies) are consistent with organizations' regulations and (2) matching service and client policies for effective service discovery are issues yet to be addressed. We thus presented a new approach (Figure 3) that allows for the automatic verification and matching of policies, using a service registry. The registry serves as a policy storage and management facility, a policy checkpoint during *service publication*, and as a policy matchmaker during *service discovery*. We extended WS-Policy with a policy conformance algorithm for policy verification at *service publication time* and used WS-Policy Intersection for policy matching at *service discovery time*. We have developed a policy information model and the policy processing capabilities for the registry. A prototype has also been implemented.

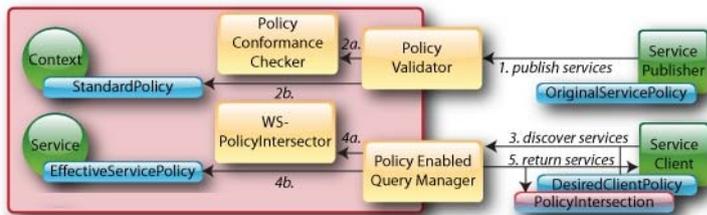


Fig. 3. A registry-centric model for policy-based service registration and discovery

4 Conclusion

We have presented in this paper a discussion about the limitations of current approaches in business-oriented design and management of qualities for SOA systems and outlined a framework for addressing this issue. Our approach is aimed at aligning business-oriented rules and requirements with system-level management via a mechanism that allows for the specification of quality-oriented business rules and

regulations and the refinement of them into system-level quality. We also provide a general mechanism that utilizes a service registry for quality-based service registration and discovery. Our future work is on monitoring techniques that can relate monitored service interactions to the original high-level business requirements and a business-oriented adaptation mechanism for overcoming non-compliance.

Reference

1. Sarbanes, P.: Sarbanes-Oxley Act of 2002. The Public Company Accounting Reform and Investor Protection Act. Washington DC: US Congress (2002)
2. Bajaj, S., Box, D., Chappell, D., Curbera, F., Daniels, G., Hallam-Baker, P., Hondo, M., Kaler, C., Langworthy, D., Malhotra, A.: Web Services Policy Framework (WS-Policy). Version 1 (2006) 2006-2003
3. Nicodemos, D., Naranker, D., Emil, L., Morris, S.: The Ponder Policy Specification Language. Proceedings of the Int'l Workshop on Policies for Distributed Systems and Networks. Springer-Verlag (2001)
4. Uszok, A., Bradshaw, J., Jeffers, R., Suri, N., Hayes, P., Breedy, M., Bunch, L., Johnson, M., Kulkarni, S., Lott, J.: KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement. IEEE 4th Int'l Workshop on Policies for Distributed Systems and Networks, 2003.POLICY 2003. (2003) 93-96
5. Phan, T., Han, J., Schneider, J.G., Ebringer, T., Rogers, T.: A Survey of Policy-Based Management Approaches for Service Oriented Systems. 19th Australian Conf on Software Engineering, 2008. ASWEC 2008 (2008) 392-401
6. Wada, H., Suzuki, J., Oba, K.: A Model-Driven Development Framework for Non-Functional Aspects in Service Oriented Architecture. 2006 Int'l Conf on Autonomic and Autonomous Systems (ICAS 2006). IEEE Computer Society, Silicon Valley, California, USA (2007)
7. Comerio, M., Paoli, F.D., Grega, S., Maurino, A., Batini, C.: WSMoD: a Methodology for QoS-based Web Services Design. Int'l Journal of Web Services Research 4 (2007) 28
8. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Journal of Network and Systems Management 11 (2003) 57-81
9. Lamanna, D.D., Skene, J., Emmerich, W.: SLAng: A Language for Defining Service Level Agreements. Proc. of the 9th IEEE Workshop on Future Trends in Distributed Computing Systems-FTDCS (2003) 100-106
10. Sahai, A., Machiraju, V., Sayal, M., van Moorsel, A., Casati, F.: Automated SLA Monitoring for Web Services. IEEE/IFIP DSOM 2002 (2002)
11. Casati, F., Shan, E., Dayal, U., Shan, M.-C.: Business-oriented Management of Web Services. Communication of the ACM 46 (2003) 55-60
12. Phan, T., Han, J., Schneider, J.-G., Wilson, K.: Quality-Driven Business Policy Specification and Refinement for Service-Oriented Systems. Int'l Conf on Service Oriented Computing (ICSOC 2008). Springer, Sydney, Australia (December 2008) 17 pages, to appear
13. Phan, T., Han, J., Schneider, J.-G., Ebringer, T., Rogers, T.: Policy-based service registration and discovery. In: Meersman, R., Tari, Z. (eds.): Int'l Conf on Cooperative Information Systems 2007 CoopIS 2007, in On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS. Springer, Villamoura, Algarve, Portugal (2007) 417-427