Non-Functional Property Driven Service Governance: Performance Implications

Yan Liu[1,2], Liming Zhu[1,2], Len Bass[3], Ian Gorton[4], and Mark Staples[1,2]

[1] NICTA, Australian Technology Park, Eveleigh, NSW, Australia
[2] School of Computer Science and Engineering, University of New South Wales, Australia
[3] Software Engineering Institute, Carnegie Mellon University, U.S.A.
[4] Pacific Northwest National Laboratory, U.S.A.
{Jenny.Liu,Liming.Zhu,Mark.Staples}@nicta.com.au,
ljb@sei.cmu.edu, ian.gorton@pnl.gov

Abstract. Service governance is a set of businesses processes, policies and technical solutions that support enterprises in their implementation and management of their SOA. The decisions of service governance, especially concerning service boundaries at the enterprise level, influence the deployment topology of business services across or within business organizations. Deployment topologies are realized by integration technologies such as Enterprise Service Bus (ESB). Service governance and technical solutions interact in a subtle way including through communication patterns and protocols between services and ESBs, as well as the deployment and configuration of ESB. These factors have a strong influence on the Non-Functional Properties (NFP) of a SOA solution. A systematic approach is essential to understand alternative technical solutions for a specific service governance decision. This paper proposes a modeling approach to evaluate the performance-related NFP impacts when mapping service governance to technical solutions using an ESB. This approach is illustrated by the quantitative performance analysis of a real world example, service governance from an Australian lending organization.

1 Introduction

Service governance is one of the key factors of achieving high quality Service Oriented Architecture (SOA). Service governance includes a set of solutions, policies and practices which enable the integration of individual business services into an enterprise level SOA. The relationship between business organization, business goals, service governance and Non-Functional Properties (NFP) is illustrated in Figure 1.

Depending on the business scenario, a business organization can have alternative choices in selecting the service boundaries. For example, a business organization with several business services implemented as web services could allocate basic processing services into branches and the individual branches forward advanced processing requests to headquarters; alternatively, individual branches could provide most of the services and only contact the headquarter services when exceptions arise. Business analysts provide input on appropriate service governance policies and the architects determine where and how to provide the services using integration technologies. These two roles must coordinate to determine the appropriate solution.

E. Di Nitto and M. Ripeanu (Eds.): ICSOC 2007 Workshops, LNCS 4907, pp. 45–55, 2009.

From the deployment point of view (see Figure 1), the boundaries of services are represented by the topology of service deployments. The deployment and integration of services across a boundary requires a service routing capability, which is a capability of an Enterprise Service Bus (ESB). An ESB is a standards-based integration platform that combines messaging, web services, data transformation, and intelligent routing in a highly distributed environment [8]. Architects make decisions on ESB topologies for the deployment of supporting services, and on the messages and event-based interactions among heterogeneous platforms hosting individual services.

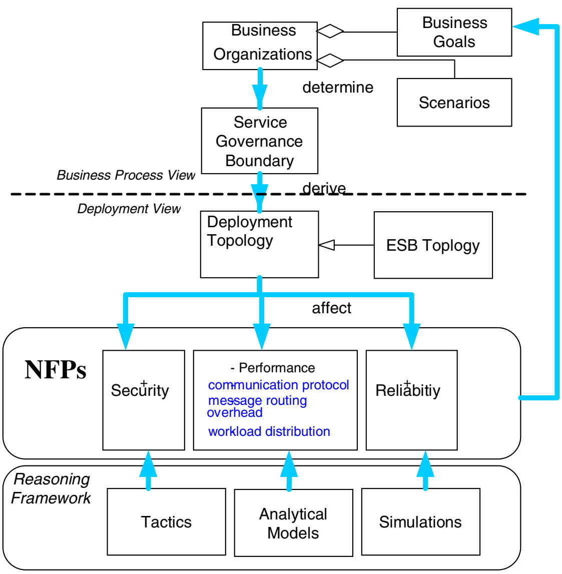


Fig. 1. Mapping between service governance and non-functional properties

The decisions about how to deploy the capabilities of an ESB to support service governance have implications on the NFPs and service level of SOA. For example, although ESBs mask the complexities of composing applications and services, they also introduce performance overheads incurred by message brokering and transformation between services and applications. Evaluating the NFPs of the ESB topology given a service governance assertion can provide architects and business analysts with quantitative feedback on whether the service governance decisions support the business goals. Architecture evaluation methods and techniques such as ATAM [5] and i-Mate [1] have been developed to evaluate the NFPs of general software architecture. Techniques such as quantitative analysis models, simulation, and architecture tactics can be used in evaluation [10]. However, these methods mainly focus on software architecture evaluation for systems supporting a single organization.

Cross-organizational evaluation, which is commonly needed in service governance using ESBs, presents new challenges for evaluating NFPs of SOA. It is not clear from the governance point of view, how technical solutions such as the use of ESB at the service boundary impact the overall performance. Moreover, the evaluation methods also need to specify and analyse the characteristics of ESBs at different levels of detail.. This is essential as business requirements can be represented at different levels of abstraction, and the evaluation results of service governance need to be interpreted according to these requirements [9]. A systematic approach is required to estimate an ESB's characteristics and to analyse the relationship between the ESB and the services it composes.

In this paper we propose a quantitative approach to evaluating the performance of ESB solutions when service boundaries are mapped to service deployment on an ESB. This approach is illustrated by constructing a performance model for a real world example, service governance from an Australian lending organization. Although this paper focuses on performance, we believe this approach can be extended to evaluate other NFPs.

2 Understanding the ESB Architecture

In order to characterize the impact of ESB topology on the performance of SOA, we first need to understand the architecture of the ESB and identify key components of the ESB that have critical impact on performance.

The capabilities of an ESB are implemented by middleware technologies such as web services, message broking, security management and so on. Despite the diversity in ESB implementations, there are common components in the architecture of an ESB. An abstract architecture of an ESB is illustrated in Figure 2.

The core functionality of an ESB is supporting disparate applications and services to communicate using different protocols as shown in the protocol stack of Figure 2. ESB message brokering provides content-based routing for dispatching a request message from one service to another service or application registered on the ESB. A service or application is connected to the service bus through an ESB 'gateway'.

A gateway (also called a proxy in some ESB implementations) can be configured with a routing table to route a request to the appropriate business service. The routing table entry defines a set of actions, including routing to a service, calling out to a service, composing messages with input parameters for invoking services, transforming output parameters to response messages, and validating the message content. At runtime the routing table guides the message flow. Each action is executed by checking the predefined expression with variables, operators and variable values associated with an action. The gateway also marshals a request from the message bus to the service.

Figure 3 shows an example using content-based routing to connect one service A to other two services B and C depending on some condition configured in the routing table encapsulated in RouteNode1. This routing configuration is attached to the ESB gateway LoanGateway1. An ESB implementation normally provides comprehensive services and utilities such as service monitoring, reporting and message security. Details of these services can be found from individual ESB implementations.

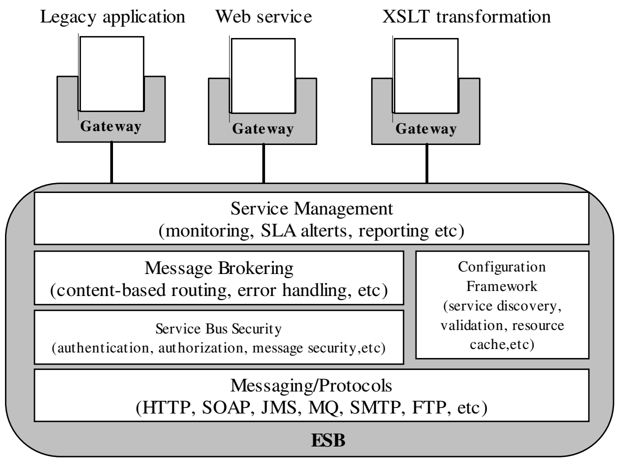


Fig. 2. High level ESB architecture

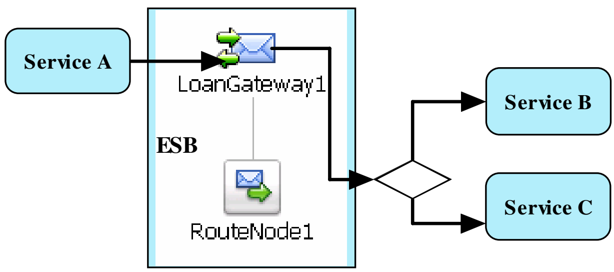


Fig. 3. Example of ESB content-based routing

The end-to-end responsiveness of a request to SOA depends on the performance characteristics of the ESB routing and the destination services. In SOAs, service applications can be considered as a composition of individual services. The problem of ensuring sufficient performance falls into two broad categories: individual services, and composite services [6]. Individual services provide service interfaces that encapsulate existing systems, so ensuring their performance necessitates managing the performance of the components, applications, and systems that lie beneath the services' abstraction. Well-established capacity planning methods, techniques and tools can be used to manage the performance of individual services. These approaches include logging-based instrumentation, and simulating the load on service interfaces by load testing in a similar way to simulating traditional web application performance.

Dealing with performance of composite services implemented using ESB is far more complex than reasoning about the performance of individual services. Services registered to an ESB can be uniquely identified by their service endpoints. An ESB composes services in a loosely coupled way by routing and transforming messages among services with different protocols. The ESB topology determines the capacity of message routing and transformation between services connected by ESB gateways.

In the rest of this paper, we propose a performance modeling-based approach that models the interactions between business services and ESB gateways in the resulting deployment architecture after the service governance assertion.

3 The Modeling Approach

Service governance decisions influence the architecture that determines the deployment topologies of services and associated technical solutions. These have implications on NFPs of the SOA. An evaluation approach that quantifies the effects of technical decisions on NFPs can provide architects a base for reasoning about their decisions for service governance. Modeling techniques have the capability of quantitatively analyzing NFPs such as performance [2][3][4]. In this paper, we propose a modeling approach to evaluate the following aspects of ESB performance:

- ESB message routing and transformation
- Protocols between ESB and business services
- Workload distribution
- Different levels of detail of ESB services

To summarize our argument, ESBs play a key role in implementing service governance decisions. The performance characteristics of an ESB are critical to the overall performance of the integrated services. The ESB performance is mainly determined by its capacity for content-based routing and message transformation. Our approach focuses on the performance overhead of ESB routing and transformation. This can simplify performance analysis and establish a base-line model, which can be further extended to incorporate more complex ESB features such as monitoring and security management.

3.1 Constructing a Hierarchical Performance Model

As we have discussed in section 2, an ESB can be introduced to integrate services within different service governance boundaries, or it can be used to connect services on the same side of a service governance boundary. The interactions between an ESB and business services should be modeled at different level of details. There are two modeling views, a system-level point of view and a component-level point of view. A system level performance model views the system being modeled as a 'black box'. This means the internal details of the box are not modeled explicitly and the box is characterized by its throughput function. A component-level model, on the other hand, takes into account the details of the resources and the way they are used by different requests. The details are explicitly considered by the model.

Our approach combines system-level and component-level views to construct a hierarchical model which is decomposable into different abstraction levels.

We consider the simple scenario in Figure 3 that an ESB is introduced to integrate business services across a governance boundary. The routing and message transformation functionalities are provided by an ESB gateway. Each business service and gateway is mapped to a resource (represented by a queue notation in Figure 4) that serves arriving requests in the analytical queuing network model. The assumption that each service

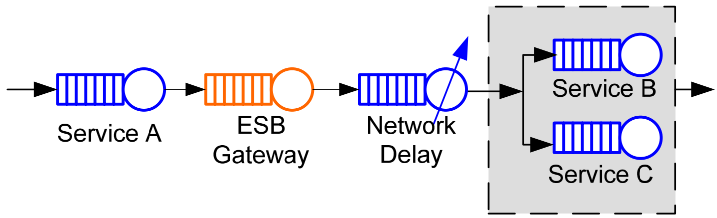


Fig. 4. Modeling interactions between ESB and business services

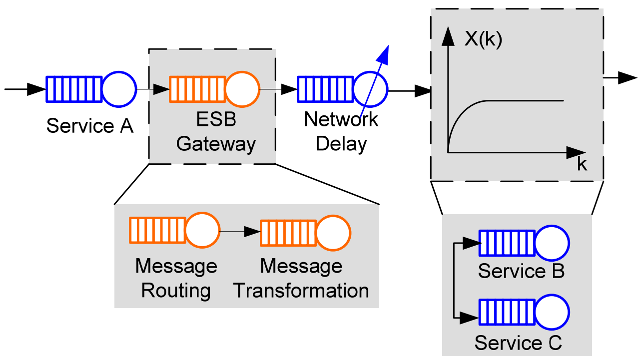


Fig. 5. Model decomposition

is load-independent simplifies the modeling complexity and reduces the effort of collecting parameter values to represent the load-dependent behavior of a service. This assumption is validated by the modeling results presented later in this section.

The connections between ESB gateways and business services can be through different types of networks, such as dedicated network channels, Ethernet, or WAN. There can be firewalls and also load balancing proxies installed along the requests invocation paths of business services. The service delay introduced by network protocols and software can be further modeled by introducing additional load-dependent queues (see Chapter 8 in [7]) and combined into the performance model as shown in Figure 4.

Notice that Service B and Service C are within the same boundary of service governance. They can be modeled as a black box resource (see Figure 5) with the throughput function defined. This throughput function can be derived using Flow Equivalence and Hierarchy Modeling techniques (see Chapter 9 in [7]) by replacing the black box with detailed models for Service B and Service C into the system model.

If the message routing or transformation functionality of the ESB requires very resource demanding computation, the deployment of these two ESB major components can be separated on different nodes to increase capacity. In the model, the queue for the ESB can be replaced by a network of queues modeling message routing and transformation, each of which has its own performance capacity.

The choice of different ESB topologies is influenced by the arriving workload for the business services. For example, in one scenario, some services might expect high volumes of traffic from a specific consuming application, while in another scenario there might be high reuse, with a service used by many different consumers. The modeling approach covers this through the flows between queues. Each flow is characterized by the arrival rate and number of requests on individual services.

This combined performance model provides the flexibility of modeling interactions between ESBs and business services at different levels of detail. The performance models illustrated in Figure 4 and Figure 5 can be applied to construct a large network of models with more than one ESB used. By manipulating the structure of the performance model, one can represent different topologies of ESB and business service governance decisions. This modeling-based approach can provide early feedback on the performance characteristics of a range of possible governance decisions.

4 An Illustrative Case: LIXI Service Governance

The modeling approach described in section 3 combines system level and component level models and enables modeling ESB topologies at different abstraction levels. We here present the results of applying it to service governance in the Australian lending industry.

Vertical industries have been developing e-business standards to improve business interoperability and streamline business services across organizational and business unit boundaries. The e-business standards in this case are essentially a set of governing rules for service. Until recently, most e-Business standards have been composed of XML-based business data models, message exchange patterns and business process models. The business governance issues and their technical implications are not well understood.

NICTA has been working with a leading Australian e-Business industry standardization body – Lending Industry XML Initiative (LIXI) that serves the lending industry in Australia. LIXI initially developed a XML-based data centric standard, later complemented by a process model described in BPMN (Business Process Modeling Notation), jointly developed with NICTA. The BPMN models now have been used as inputs to developing a reference implementation using web services. However, the same BPMN model may imply different organizational and business unit boundaries in different scenarios. This consequently has technical consequences in the reference implementation. In this paper, we use examples from the LIXI domain as an illustrative case study to demonstrate our approach.

Figure 6 shows a simplified lending business process from LIXI. A loan application is received and then initially validated. Then a loan process service will further process the loan application based on some business rules, contact a credit bureau and verify physical evidence. If an exception occurs, the loan application will be forwarded to an authority service for special consideration. When a decision is finally made, some service will be responsible for contacting the original applicant. The service here could be a human or an automated web service depending on the sophistication of participating parties. The services could reside within the same governance boundary (in one business unit) or could be spread across different business units or even different organizations. We have indicated four possible business unit/organization in Figure 6 using dotted lines. Here, we illustrate two real world scenarios from the LIXI domain:

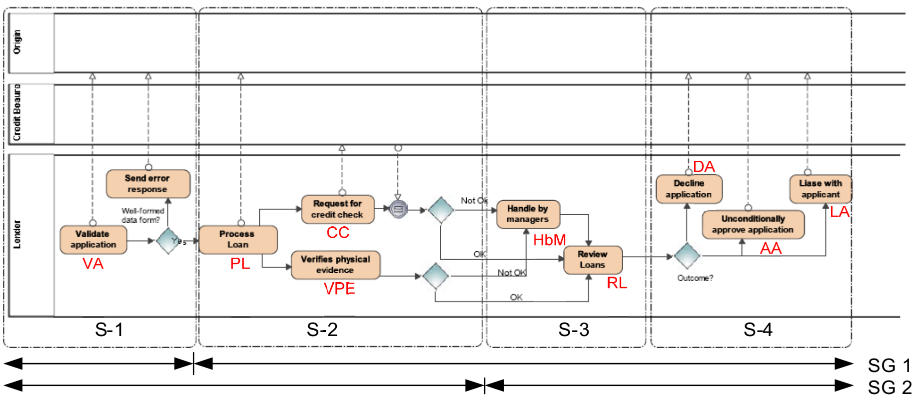


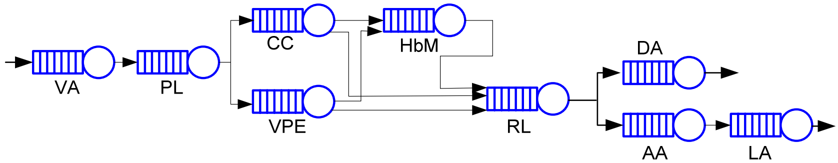
Fig. 6. BPMN model for loan application

Bank Branch – Headquarter Scenario: In this scenario, one business unit is 1 representing a branch of a bank. The other business unit is 2,3,4 representing the headquarter. A branch receives a loan application and only does minimal processing such as paper work, data entry and validation. It then forwards the application to the headquarters for further processing. The business governance boundary would be around branch and headquarter respectively.

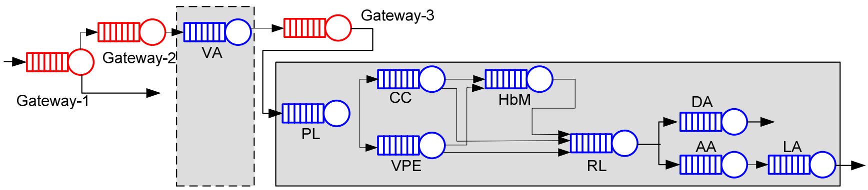
Mortgage House – Bank Scenario: In this scenario, one organization is 1 and 2 representing a mortgage house. A mortgage house is usually a reseller of loan products from a bank. The other organization is 3 and 4, representing a bank. A mortgage house will thoroughly process a loan application first and only contacts the bank service if an exception arises. The business governance boundary would be around the mortgage house and the bank respectively.

Depending on the network links, communication protocols and service gateways between these business units and organizations, the deployment topology and technical implications could be very different. In order to explore the performance implications of the above two scenarios, we constructed performance models that represent the topologies and the factors associated with each topology.

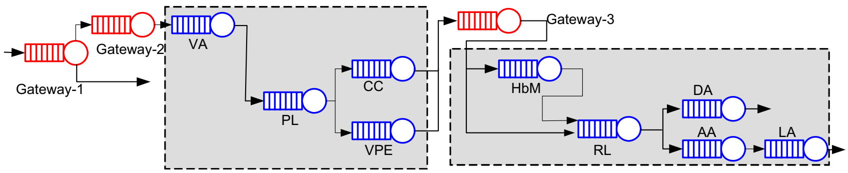
We first constructed the baseline model with only business services involved, and then we extended this model with ESB gateways to represent the governance scenarios. The baseline model is shown in Figure 7 (a) with each queue representing a unit business service. The performance models for the headquarters and bank scenario are constructed in Figure 7 (b) and (c) respectively. The assumption is that the ESB gateways are introduced at the boundary of services to support the governance decision. Business services within the same service governance boundary are grouped together into a black box system level resource (covered by the gray dashed boxes), whose performance characteristics are presented as the overall throughput of these services. As we discussed in section 3, each system level black box resource can be further decomposed into networks of queues, and solved to derive the throughput function.



(a) The base line performance model of business services



(b) The performance model for service governance scenario one (SG1)



(c) The performance model for service governance scenario two (SG2)

Fig. 7. The performance models of two service governance scenarios

The models shown in Figure 7 still have simplified assumptions of the real world deployment of these services. The protocols that connect ESB gateways and business services are not modeled explicitly. In reality, if the network overhead is not insignificant, extra queue should be introduced to model this factor.

Further refinement of the models involves planning the capacity of the ESB, and deciding whether the message routing and message transformation should be deployed into separate nodes.

For demonstration purposes, we simplify our models by assuming that the set of services on S-1, S-2, S-3 and S-4 are each deployed on their own machines. For both scenarios, each ESB gateway is deployed on a separate machine. The service demand and request arrival rate are estimated by NICTA's architects who lead the design and implementation of LIXI web service infrastructure. The network work delay D is simply modeled as a function of bytes sent and received B, and the network speed S, that is $D= B/S$. We solve each model by using the Mean Value Analysis (MVA) solution algorithms (see Chapter 8 in [7] for details).

The estimated results are shown in Figure 8. As the services are deployed on the same machines for both scenarios, the performance difference mainly results from the higher service demand on ESB gateway 3 as this gateway is responsible for routing two services with more messages and larger size of each message. The estimation demonstrates that these two scenarios reveal comparable performance with difference around 5 to 9 percentage points.

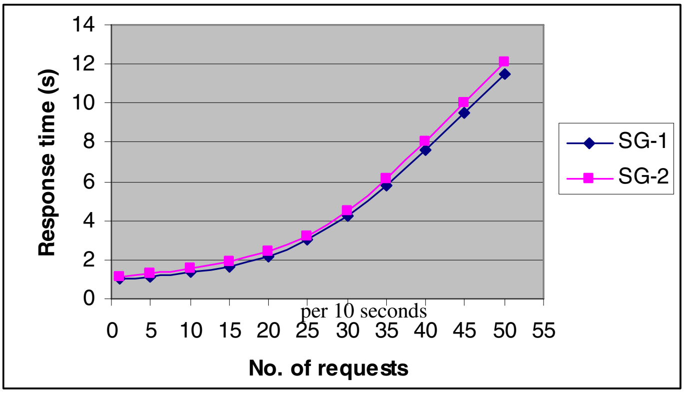


Fig. 8. Performance evaluation of ESB topologies

Based on the assumption that the hardware and software capacity are identical in the two scenarios the analysis shows that the performance distinction is not significant. However, in the reality, the physical deployment could cause variations of capacity for each scenario, leading to different performance results.

5 Conclusion

The interactions between service governance and service integration solutions have effects on the NFPs of SOA. A quantitative evaluation of the technical solutions can help architects quantify the decisions of service governance. This paper proposes a modeling approach to evaluating the performance impact when mapping service governance to technical solutions using Enterprise Service Bus (ESB). The modeling approach combines two levels of performance models and provides the flexibility of modeling the interactions between ESB gateways and business services in different level of abstraction. This approach is illustrated by a case study of LIXI loan application.

This LIXI application provides two scenarios for service governance and demands using ESB for service integration across governance boundary. ESB's core features including content-based routing and message transformation are applied as technical solutions to support service governance. The performance evaluation of these two ESB deployment scenarios follows our combined modeling approach and produces preliminary results for understanding the performance characteristics of the LIXI loan application.

This case study focuses on the performance modeling method instead of doing performance evaluation of the ESB with stress testing. Therefore, the workload imposed is synthetic in section 4. For a real world application, the workload characterization approach described in [7] can be applied. Obtaining the parameter values to solve the model requires the measurement of the delay incurred by individual ESB infrastructure gateway/proxy service. This can be achieved using performance testing tools.

In this paper our modeling approach focuses on the performance aspect. In the future, this approach will be further extended to explore the implication on other NFPs when using ESB to support service governance.

References

[1] Liu, A., Gorton, I.: Accelerating COTS Middleware Acquisition: The i-Mate Process. IEEE Software 20(2), 72–79 (2003)

[2] Almeida, V.A., Menascé, D.A.: Capacity Planning: An Essential Tool for Managing Web Services. IT Professional 4(4), 33–38 (2002), http://dx.doi.org/10.1109/MITP.2002.1046642

[3] Chen, S., Yan, B., Zic, J., Liu, R., Ng, A.: Evaluation and Modeling of Web Services Performance. In: Proceedings of the IEEE International Conference on Web Services (ICWS 2006), September 18 - 22, 2006, vol. 00, pp. 437–444. IEEE Computer Society, Washington (2006), http://dx.doi.org/10.1109/ICWS.2006.59

[4] Catley, C., Petriu, D.C., Frize, M.: Software Performance Engineering of a Web service-based Clinical Decision Support infrastructure. In: Proceedings of the 4th International Workshop on Software and Performance, WOSP 2004, Redwood Shores, California, January 14 - 16, pp. 130–138. ACM Press, New York (2004), http://doi.acm.org/10.1145/974044.974066

[5] Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice, 2nd edn. Addison Wesley Professional, Reading (2003)

[6] Menascé, D.A.: QoS Issues in Web Services. IEEE Internet Computing 6(6), 72–75 (2002), http://dx.doi.org/10.1109/MIC.2002.1067740

[7] Menascé, D.A., Almeida, V.A.: Capacity Planning for Web Services: metrics, models and methods. Prentice Hall, Englewood Cliffs (2001)

[8] Keen, M., et al.: Patterns: Implementing an SOA Using an Enterprise Service Bus. IBM redbook (2004), http://www.redbooks.ibm.com/abstracts/SG246346.html, ISBN 0738490008

[9] O'Brien, L., Merson, P., Bass, L.: Quality Attributes and Service-Oriented Architectures (CMU/SEI-2005-TN-014). Software Engineering Institute, Carnegie Mellon University, Pittsburgh (2005)

[10] Tran, P., Greenfield, P., Gorton, I.: Behavior and Performance of Message-Oriented Middleware Systems. In: ICDCS Workshops, pp. 645–654. IEEE Computer Society, Los Alamitos (2002), http://csdl.computer.org/comp/proceedings/icdcsw/2002/1588/00/15880645abs.htm