

Risk Management for Service-Oriented Systems

Natallia Kokash

DIT - University of Trento, via Sommarive, 14, 38050 Trento, Italy
natallia.kokash@dit.unitn.it

Abstract. Web service technology can be used for integrating heterogeneous and autonomous applications into cross-organizational systems. A key problem is to support a high quality of service-oriented systems despite vulnerabilities caused by the use of external web services. One important aspect that has received little attention so far is risk management for such systems. This paper discusses risks peculiar for service-based systems, their impact and ways of mitigation. In the context of service-oriented design, risks can be reduced by selection of appropriate business partners, web service discovery, service composition and Quality of Service (QoS) management.

Advisors. Vincenzo D'Andrea.

1 Introduction

Rapidly evolving web service technology is a key mechanism for simplifying large-scale business operations by consumption of ready-to-use services. More and more software is becoming available as web services, loosely-coupled functional entities accessible via well-defined user interfaces. These interfaces are published in registries and can be discovered by potential customers. Several web services from different providers may have to be integrated to implement real-world business information systems. Such systems, in their turn, can be available as web services for invocation by end-users or further integration.

Web service implementation details are normally hidden. Their potential clients reason about service functionalities being unaware of their internal structure. This spawns a significant challenge for those who want to use existing services as parts of new web systems. For attracting customers, these systems must be responsive, robust, and always available. They should support concurrency demands and deal gracefully with load variations. Such requirements are commonly referred to as Quality of Service (QoS). Issues related to QoS support for composite web services has found considerable research interest.

A composite web service is reactive to any changes in the behavior of constituent services. Service-Oriented (SO) systems are subject to risks caused by architectural vulnerability (e.g., technical problems, security-level threats) and by conflicting interests of the involved partners. A mechanism to support designers of service-based applications in vulnerability assessment of their systems is needed. A way of gathering the requisite data to make a good business or

technical judgement is provided by *risk management*. Among the main steps of risk management are *risk identification* (surfacing risks before they become problems), *risk analysis* (converting identified risk data into decision-making information), and *risk control* (monitoring the status of risk and actions taken to mitigate them). Regarding SO design, risk management can be applied to decide whether to entrust a part of functionality to an existing web service, which of the existing services to choose, how to protect data exchanged between partners, which additional controls must be implemented, how many alternative services are required for a particular task, and so on.

The rest of the paper is organized as follows. Section 2 provides main definitions and basic information about risk management. Section 3 discusses the distinctive features of risk analysis for SO systems. In Section 4, a high-level framework for design and reconfiguration of SO systems is proposed. The last section concludes the paper and outlines future work.

2 Software Risk Management

Risk management operates with notions of *assets* (objects of the protection efforts such as system components or data), *threats* (danger sources), *vulnerabilities* (defects in system design, implementation, or internal controls), threat *probabilities* (likelihoods that given negative events will be triggered), and their *impacts* on the organization (tangible or intangible, e.g., monetary loss or breach of reputation, law, regulation, or contract). In order to mitigate risks, *countermeasures* (management, operational, and technical controls that adequately protect the system) are applied [1].

Risk r is defined as a probability p of a threat e multiplied by a respective magnitude q of its impact: $r(e) = p(e)q(e)$. Real-world systems are unlikely to have a single risk factor. Risk of a set of independent threats can be calculated as a sum of risks for each particular threat [2]. In more general case, analysis of conditional dependencies among threats (their probabilities and impacts) is required. Often probability calculation and impact estimation are extremely rough, but still help to handle technical vulnerabilities of the system.

The goal of risk management frameworks [1][3] is to help designers to manage software projects within established time and budget constraints. Several works examine risk management for business processes [4][5]. A *business process* is a structured set of activities designed to produce a specified output for an organization or a consortium. Business processes are subject to errors in each of their components: to enable the successful completion of a business process it is important to manage the risks associated with each sub-activity. A company can rely on someone else to run certain business functions. A survey of current practices in risk management for project *outsourcing* (a formal agreement with a third party to perform a service) can be found in [6].

There exist a bulk of potential threats for cross-organizational systems. In the same time, there can be no stakeholders with a full knowledge about the system. Among the most common risk factors are tasks involving third parties, use

Table 1. Risks in service-oriented applications

Event	Assessment	Mitigation
Loss of service	Likelihood and implications of service unavailability and network-related problems	Use for non-critical tasks and tasks that can be suspended. Use of alternative services for critical tasks. Establishment of trusted relations and service level agreements. Redesign (own code or locally deployed components).
Loss of data	Likelihood and implications of data loss	Data replication. Exchange of non-critical data. Establishment of trusted relations. Service level agreements.
Loss of users	Analysis of user expectations.	Warn users about possible problems. Provide different quality layers. Collect feedback from users.
Unexpected service behavior	Analysis of consistency and completeness of published service specifications.	Service testing. Clarification of the specifications. Service level agreements.
Specification changes	Evaluation of implications of format changes and loss of service.	Use of services on a small-scale and plan for redesign. Discovery of alternative services.
Performance problems	Service testing. Likelihood and implications of inadequate service performance	Service performance monitoring. Use for non-critical tasks. Service level agreements.
Lack of interoperability	Likelihood and implications of application lock-in and integration loss.	Evaluation of integration capabilities. Use of alternative services. Redesign.
Contract violation	Reputation of a service.	Run-time monitoring. Use of web services for non-critical tasks. Use of alternative services for critical tasks.

of unfamiliar or emerging technologies, organizational problems, unclear goals, absence of quality controls and effective management. Being a state-of-the-art in business process integration techniques, SO systems are subject to the mentioned risks as well.

3 Risk Management for Service-Oriented Systems

Among a set of risks peculiar for SO systems are risks caused by service providers (disposal of a service, changes in interface and behavioral logics of a service, contract violation, obstruction of a new contract with worse conditions, disclosure of user data) and technical aspects (network or service failures, problems with semantic interoperability). Table 1 summarizes some common threats for service-based applications. Security threats (information disclosure, spoofing and tampering, downgrade, repudiation, denial of service, etc.) and methods for their prevention are discussed in WS-Policy and WS-Security specifications.

Techniques for risk management fall into five categories: (i) *risk avoidance* involves altering the original system design to remove particularly risky elements; *risk reduction* employs methods that reduce the probability or impact of a risk occurring (e.g., use of alternative services to reduce the probability of a service loss, data replication to avoid a loss of data); (iii) *risk transfer* moves the ownership of the risk to a third party by contract (e.g., contracts stipulating penalties to web services for information disclosure); (iv) *risk deferral* entails deferring

decisions to a date when a risk is less likely to happen or less severe; (v) *risk retention* implies that certain risks have to be accepted.

For every accepted risk, a designer must put controls in place that detect the corresponding event when it triggers. The idea of automatic service composition implies that service-based applications are created without (or with minimum of) human intervention: according to the principle of *late binding*, abstract service descriptions are mapped to real services on demand. Following this trend, a tool for automatic risk analysis should be developed.

4 Risk Management Framework

To enable design of reliable SO systems, an appropriate methodology and supporting tools are needed. It is the matter of risk analysis to decide whether a service-oriented computing can be applied for some business application. Zimmermann et al. [7] identify risk mitigation strategies before kicking off any premature implementation work as one of the most important lessons learnt from the project involving SO architectures, process choreography, and web services in the telecommunications industry. In our vision, risk management is a context sensitive and iterative process. Organizations have to perform systematic risk assessment both for systems being developed and for deployed systems that support runtime reconfiguration [8].

The proposed risk management framework is shown in Fig. 1. Once the business process has been modelled at the abstract level, the further analysis is needed to gather information about potential business partners and useful web services. Principal users and their expectations should be examined and QoS information about services acquired. The latter can be provided by service owners, other clients or certification agencies. Then, the analysis of system artifacts is performed. In particular, web services discovered at the previous step must be tested and their conformance with system requirements, both functional and non-functional, assessed. At the next step, the initial system configuration is selected or the system is reconfigured. Among possible actions are: *service replacement*, which means that a component of a composite web service is changed; *structural change*, which means the systems logical structure is changed; and *geometrical change*, which means that the logical application structure remain fixed, but the physical structure (hardware, network topology, communication protocols, etc.) is changed [8]. Risks for the chosen configuration are identified using a predefined ontology of threats affecting the system.

Probabilities of threats can be estimated quantitatively based on monitored data. For example, probability of a loss of service is defined by service unavailability rate, percentile of incorrect past executions to total number of invocations defines probability of unexpected service behavior, normalized incompliance between a required web service and a discovered one can be seen as a probability of a lack of semantic interoperability. For assessing an impact of each particular threat on the system, dependency diagrams, which outline interrelationships of various functional elements, can be involved. The information about

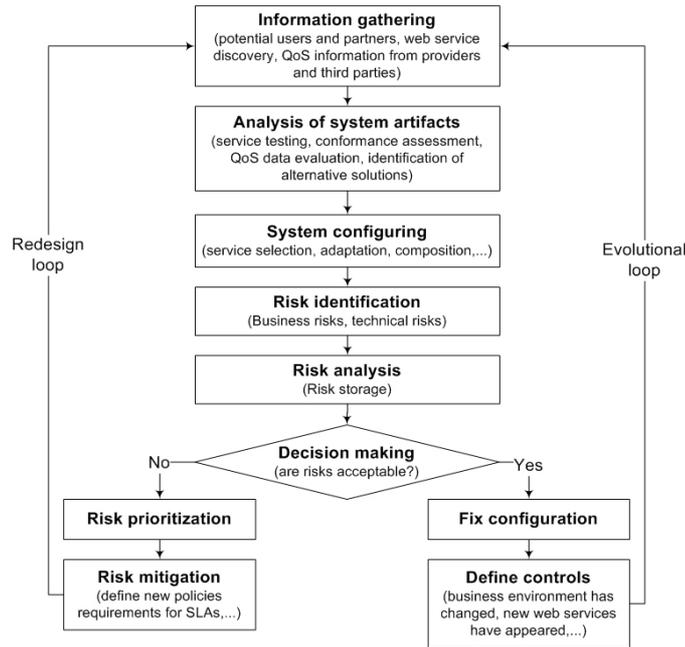


Fig. 1. Risk management framework for service-oriented systems

risk assessment (unique ID, description, reason, warning signs, probability, impact, timescale, cost of mitigation, expected level of risk after mitigation, etc.) is stored to allow for the comparative analysis of different system configurations.

If the overall risk level is not acceptable and/or there exist a potential for its mitigation, the designers must try to reduce it. Risk prioritization implies that the identified risks can be segmented into categories (high, medium, and low) and managed accordingly. Risk mitigation can be achieved either through system reconfiguration or through contract management. New functional requirements, such as necessity of data encryption, additional controls, etc. can be unveiled. Finally, if the overall risk is acceptable, the system configuration is fixed and the service-based application can be exploited. Predefined controls can detect important changes in the system or in the business environment and initiate an evolutionary loop when some parts of the system are replaced or adopted to meet new business requirements.

5 Conclusions and Future Work

Risk management is an essential component for SO design affecting structure of a system, eliciting requirements for service discovery, selection and contracting. It can be partially automated to enable design of reliable service-based systems using automatic service composition techniques.

In our recent work [9] a detailed description of risk-based service evaluation and selection strategies is given. In [10] an extensible monitoring service able to collect client reports on service invocations and suggest reliable services to users with similar needs is presented. It provides a basis for gathering QoS statistics (e.g., types of possible web service exceptions) and aims at improving quality of web service discovery [11]. This will allow for the easier integration of existing functionalities into new software applications.

In our future work we are planning to elaborate the above ideas, supplementing design of service-based business processes with ability to model risks and infer necessary preventive strategies and controls.

References

1. Verdon, D., McGraw, G.: Risk analysis in software design. *IEEE Security and Privacy* (2004) 33–37
2. Roy, G.G.: A risk management framework for software engineering practice. In: *Australian Software Engineering Conference (ASWEC)*, IEEE Computer Society (2004) 60–67
3. Freimut, B., Hartkopf, S., Kaiser, P., Kontio, J., Kobitzsch, W.: An industrial case study of implementing software risk management. In: *ESEC/FSE*, ACM Press (2001) 277–287
4. zur Muehlen, M., Rosemann, M.: Integrating risks in business process models. In: *Australasian Conference on Information Systems (ACIS)*. (2005)
5. Neiger, D., Churilov, L., zur Muehlen, M., Rosemann, M.: Integrating risks in business process models with value focused process engineering. In: *European Conference on Information Systems (ECIS)*. (2006)
6. O’Keeffe, F., Vanlandingham, S.: Managing the risks of outsourcing: a survey of current practices and their effectiveness. White paper, Protiviti, http://www.protiviti.com/downloads/PRO/pro-us/product_sheets/business_risk/Protiviti ORM WhitePaper.pdf (2004)
7. Zimmermann, O., Doubrovski, V., Grundler, J., Hogg, K.: Service-oriented architecture and business process choreography in an order management scenario: Rationale, concepts, lessons learned. In: *Conference on Object-oriented Programming, Systems, Languages and Applications (OOPSLA)*, ACM Press (2005) 301–312
8. Li, Y., Sun, K., Qiu, J., Chen, Y.: Self-reconfiguration of service-based systems: A case study for service level agreements and resource optimization. In: *Int. Conference on Web Services (ICWS)*, IEEE Computer Society (2005) 266–273
9. Kokash, N., D’Andrea, V.: Evaluating quality of web services: A risk-driven approach. In: *International Conference on Business Information Systems (BIS)*. Volume 4439 of LNCS., Springer (2007) 180–194
10. Kokash, N., Birukou, A., D’Andrea, V.: Web service discovery based on past user experience. In: *International Conference on Business Information Systems (BIS)*. Volume 4439 of LNCS., Springer (2007) 95–107
11. Kokash, N., van den Heuvel, W.J., D’Andrea, V.: Leveraging web services discovery with customizable hybrid matching. In: *International Conference on Service-Oriented Computing (ICSOC)*. Volume 4294 of LNCS., Springer (2006) 522–528