



High-Assurance Service-Oriented Architectures

Jing Dong, University of Texas at Dallas

Raymond A. Paul, US Department of Defense

Liang-Jie Zhang, IBM T.J. Watson Research Center

Assuring multiple quality attributes—such as reliability, availability, performance, security, and real-time response—for a variety of critical applications makes it essential to develop practical techniques for implementing high-assurance service-oriented systems.

Recent advances in service-oriented technologies allow registering, discovering, requesting, and providing software services online. Such loosely coupled and independent software services form a service-oriented architecture (SOA) with the support of network resources, and users can access them without knowledge of their underlying platform implementation. Service computing has permeated many aspects of modern society—from planning, decision-making, and services industry modernization to multimedia interactive interfaces. Many mission-critical systems, such as healthcare, traffic control, and defense systems, also apply service-oriented technologies.

Despite this transition from traditional computer-based systems to a service-oriented paradigm, the high-assurance requirements of service-oriented critical systems remain the same. These systems still must be highly reliable, secure, available, timely, fault-tolerant, and dependable. Unfortunately, it might be even harder to achieve these high-assurance requirements for service-

oriented systems than for traditional computer-based systems because, unlike traditional systems, services can be designed in-house or delivered by third parties.

Additionally, advances in networking infrastructure have enabled wide-area interaction and telecontrol that has increased the importance of secure communication and online access to remote systems. These concerns for assuring multiple quality attributes—such as reliability, availability, performance, security, real-time response, and so on—for a variety of critical applications make it essential to develop practical techniques for implementing high-assurance service-oriented systems.

In the past, practical methods of developing high-assurance systems have mostly focused on computer-based techniques. However, existing methods for delivering high-assurance systems might not be readily applicable in a service-oriented environment.

There is no standard way to define high-assurance properties in service specifications at this time. Service interfaces normally focus on the descriptions of functional aspects, such as input, output, preconditions,

and effects (IOPE). The high-assurance properties of a service are generally unclear or are defined in an ad hoc manner in the service interfaces. This poses new challenges for service discovery with high-assurance requirements.

In addition, service compositions—especially dynamic compositions in the context of solutions, applications, and services such as IT or business-process outsourcing—introduce new problems. Not only do the IOPE service parameters need to match one another, but the overall system requirements also need to be satisfied. The service composition workflows can be complex—making the analysis of high-assurance properties in these workflows even harder. Thus, adapting existing methods—while simultaneously developing new approaches—will aid the transition to service-oriented high-assurance systems.

IN THIS ISSUE

This special issue explores the latest research in high-assurance SOAs. The response we received to our call for papers was considerable, with the sources including diverse academic organizations, government agencies, and industry. Unfortunately, we can only include five articles describing different aspects of high-assurance SOAs.

In “CROWN-C: A High-Assurance Service-Oriented Grid Middleware System,” Paul Townend and colleagues tackle the new challenges introduced by service orientation in high-assurance systems. There is an implicit distrust of services from third parties. The authentication between the users and services from different security domains might not apply the traditional access control methods. CROWN-C is a grid middleware system that supports the development and assessment of service-oriented systems with high dependability and security. It supports fault tolerance, fault-injection-based assessment facilities, multiparty authentication capabilities, and automated trust negotiation.

In “On Testing and Evaluating Service-Oriented Software,” Wei-Tek Tsai and colleagues investigate the problem of how to efficiently test a group of services that implement the same specification. Their group testing approach can verify both atomic and composite services through a voting agent that generates an oracle for each test case based on the majority principle. In addition, the test cases are ranked based on their coverage relationship for efficient selection.

When different Web service QoS parameters are configurable at runtime, the composition decision becomes much more complicated, as it must determine individual Web services as well as their configurable parameters in the context of the overall system. In “QoS-Reconfigurable Web Services and Compositions for High-Assurance Systems,” I-Ling Yen and colleagues discuss the reconfigurability of QoS requirements for Web services and their composition. To solve this prob-

lem, they present a rule-based reconfigurable service transformation and QoS analysis method.

The scheduler needs to detect a service failure quickly so that it can reassign work units in a timely fashion. In “Assuring Timeliness in an e-Science Service-Oriented Architecture,” John Sloan, Taghi Khoshgoftaar, and Venkat Raghav target the dynamic scheduling of services with time constraints. The decision-making process is modeled formally in Petri nets and timed automata. A model checking and a simulation environment can verify real-time systems.

In “Specifying High-Assurance Services,” Colin Atkinson and colleagues advocate a more human-readable method for service specification that is amenable to build-in test. Their specification is based on UML/OCL and uses test cases based on tabular documents. It defines the desired relationships not only between system inputs and outputs but also between the parameters and results of different operation invocations. Their specification method can help validate that the right services are being selected.

We hope you enjoy the selected set of articles. The standardization of high-assurance aspects of individual services as well as solution/system-level services will help further the development of high-assurance SOAs. ■

Jing Dong is an assistant professor in the Department of Computer Science at the University of Texas at Dallas. His research interests are high-assurance system engineering, formal methods, software engineering, and services computing. Dong received a PhD in computer science from the University of Waterloo. He is a member of the IEEE and the ACM. Contact him at jdong@utdallas.edu.

Raymond A. Paul is the technical director for the C2 Policy Directorate at the US Department of Defense Office of the Secretary of Defense Networks and Information Integration. His research interests include reliability, security, and vulnerability of computer systems and mission-critical applications. Paul received a PhD in software engineering. He is a Fellow of the IEEE Computer Society and a member of the ACM. Contact him at raymond.paul@osd.mil.

Liang-Jie Zhang is a research staff member and program manager of application architectures and realization at IBM T.J. Watson Research Center. His research interests include SOA reference architectures and design platforms. He is the editor in chief of the IEEE Transactions on Services Computing. Zhang received a PhD in pattern recognition and intelligent control from Tsinghua University. He is a senior member of the IEEE. Contact him at zhanglj@us.ibm.com.