Editorial Preface

A Framework to Ensure Trustworthy Web Services

Jia Zhang, Northern Illinois University, USA
Liang-Jie Zhang, IBM T.J. Watson Research Center, USA

BACKGROUND

Simply put, a Web service is a programmable Web application that is universally accessible through standard Internet protocols. The paradigm of Web Services has been changing the Internet from a repository of data into a repository of services along the following three dimensions: (1) facilitating business-to-business (B2B) collaboration, (2) increasing cross-language and cross-platform interoperability for distributed computing and resource sharing over the Internet, and (3) opening a new cost-effective way of engineering software to quickly develop and deploy Web applications. Therefore, the paradigm of Web Services is considered to be the model of Internet computing for the future.

However, the adoption of Web Services in industry is actually quite slow. One of the essential reasons is because of software trustworthiness, which is coined to represent people's confidence in software products (Parnas et al., 1990). At the present time, software trustworthiness, or so-called trustworthy computing[1], is considered extensively to be the paramount factor that decides the success of a software product (Mundie et al., 2004). However, it is not clear yet how this new model of Web Services ensures any measurable software quality. As a result, many companies are reluctant to employ Web Services to conduct their business.

In reality, the technology of Web Services is still in its infancy, while the trustworthiness, except the security, of Web Services-oriented computing has not gained significant attention. The community is currently preoccupied by low-level technical mechanisms of implementing Web Services (e.g., how to publish a Web service, how to compose Web Services, what is the overall architecture of Web Services-oriented system, etc.).

The Web Services community has been putting significant efforts on offering some promise to address the security challenges related to Web Services. WS-Security (2004) standard was proposed as a family of protocols that enhances the messaging technique to solve three basic problems about the quality of protection of Web Services: authentication and authorization of users, message integrity, and message encryption. Focusing on secure communication, these mechanisms can be used to accommodate a wide range of security models and encryption technologies. The WS-Security's six enhanced models, as shown in Figure 1, are proposed to help establish secure interoperable Web Services (WS-Security, 2004): (1) WS-Policy, (2) WS-Trust, (3) WS-Privacy, (4) WS-Authorization, (5) WS-SecureConversation, and (6) WS-Federation. However, WS-Security and related techniques and languages only address the security issue of Web Services-centered computing, while trustworthiness is a

holistic property that encompasses many more attributes beyond security, such as reliability, safety, survivability, interoperability, availability, fault tolerance, performance, and so forth (Neumann, 2004).

WS-TRUSTWORTHY

The techniques that ensure trustworthy Web Services are beyond the current state of the art. However, this is exactly the time when the trustworthy issues should be raised, so that they do not need to be grafted on later. The rationale is apparent; it is preferable to build in features like trustworthiness ahead of time rather than trying to retrofit them later.

Our envision to tackle this trustworthy Web Services issue is a new layer on top of the current Web Services framework, as illustrated in Figure 1. The current status is at the layer of WS-Security, with the six enhanced models prepared. A novel layer, which we call WS-Trustworthy, should be introduced in order to promise trustworthy Web Services to the applications to be built.

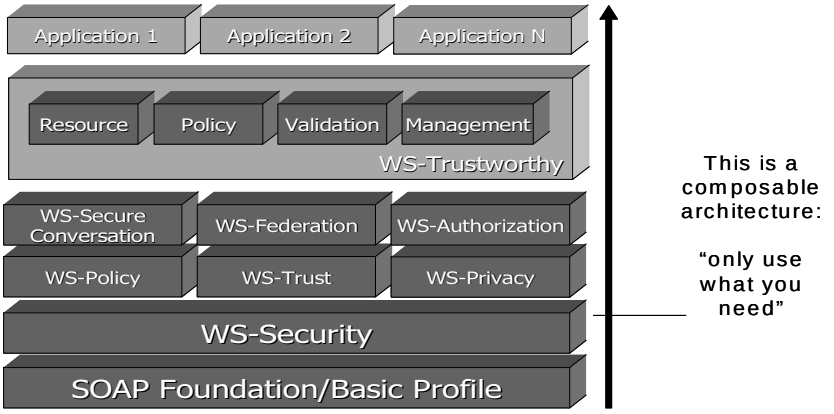
We believe that four key elements are imperative to safeguard trustworthy Web Services computing: resources, policies, validation processes, and management.

- **Resources:** The process of computing involves different categories of participant entities, such as the organizations, users, and engineers who engage in the life cycle of a Web service design and development by acting in different roles (e.g., developers, testers, analysts, managers, etc.) and other entities (e.g., agents, if agents' technology is adopted). Every entity needs to take responsibility to assure the trustworthiness of the software project. Different roles and their responsibilities need to be identified and clearly delineated.
- **Policies:** Policies identify the factors that are likely to compromise the trustworthiness; in other words, what constitute trustworthiness or how these factors can best be measured. Policies should also explicitly address roles and their responsibilities and expected behaviors.
- **Validation Processes:** Software trustworthiness is normally defined as a combination of a set of software attributes, or so-called ilities: reliability, security, safety, maintainability, survivability, availability, testability, interoperability, performance, fault tolerance, and so forth (Neumann, 2004). Trustworthiness control involves addressing each of these factors. These are the procedures that document how policy objectives are to be achieved and verified.
- **Management:** Trustworthiness should be traced and monitored as a programmatic entity throughout the whole life cycle of a Web Services-centered project.

Based upon these four elements, a componentized framework is proposed to assure trustworthy computing in the domain of Web Services. The framework is composed of four trustworthiness assurance components: resource model, policy model, validation process model, and management model. Meanwhile, this framework is considered to be oriented to Web Services-centered computing in the sense that each model is equipped with an ad hoc Web Services language or standard.

1. **Trustworthy Resource Model:** Different roles are identified in this model to represent different categories of entities that are involved in the computing; each has its dedicated responsibility to assure trustworthy computing. This model will facilitate a role-based trustworthiness assurance. We envision that a set of basic roles will be predefined, such as organization, user, role player, and other entities.

Figure 1. Envisioned trustworthy Web Services framework



- **Organization:** This resource refers to both the organizations that are involved with the application system and the ones that provide Web Services as components.
- **User:** This resource refers to the users of the application system.
- **Role Player:** This resource refers to the people who engage in the software life cycle by acting in different roles (e.g., developers, testers, analysts, etc.).
- **Other Entity:** This resource refers to other entities involved. For example, if the agents technology is adopted, agents are introduced into the system; thus, agents should be identified as resources.

We propose that roles can be formally defined using ad hoc WS-Resources (WS-Resources) language, which was proposed to facilitate the universal access of stateful resources contained in Web Services, due to three characteristics that roles possess: (1) uniqueness — each role has a distinguishable identity and lifetime; (2) statefulness — each role maintains a specific state that can be materialized using XML; and (3) accessibility — the information of each role should be accessed through one or more Web Services to provide another dimension of trust.

2. **Trustworthy Policy Model:** Policies should explicitly address roles and their responsibilities and expected behaviors. Policies identify the factors that are likely to compromise the trustworthiness. Trustworthiness control involves addressing each factor. However, there does not appear to be a clear consensus in practice or in the literature as to what constitutes trustworthiness or how these factors can best be measured.
   It should be noted that the policy model does not contain detailed technical information. For example, a policy may require that all SOAP messages sent to a service provider over the Internet be protected. How to realize this policy is a validation process issue, though, whether the SOAP message will be encrypted first before being sent to the service provider, or whether the SOAP message will not be encrypted but will be sent through an encrypted channel. For each Web service, a set of high-level policies should be predefined in this layer, such as security, reliability, safety, survivability, and so forth. WS-Policy (WS-Security) can be utilized to define the policies.
3. **Validation Model:** Validation processes are meant to provide reasonable assurance

that the system of trustworthiness control is relevant, adequate, and complied with in practice. Validation processes normally include the development of a general strategy and the preparation of a detailed approach to the corresponding policies and may also outline the supervision and review responsibilities and other trustworthiness control procedures specific to the trustworthiness requirement.

Compared to the policy layer, the validation process model is less stable. It is unlikely that the policies will change radically oftentimes. On the other hand, however, due to the ever-evolving technologies and products, the validation process layer may change on a regular basis to adapt to new technological changes. It should be noted that the different validation processes that are associated with the same policy should achieve the same objective.

As a language that can be used to specify business processes and business interaction protocols, BPEL4WS (2003) can be used to model the validation process.

4. **Management Model:** This model intends to monitor and track the application of trustworthiness control policies and procedures to obtain reasonable assurance that the system of trustworthiness control is suitably designed and effectively applied. Monitoring involves an ongoing consideration and evaluation of (1) the relevance and adequacy of the trustworthiness control policies and validation procedures, (2) the appropriateness of the resources provided, (3) compliance with trustworthiness control policies and validation procedures, and (4) the consistency of the policies and validation procedures with the developments.

BPEL Integration Development Environment (IDE), such as Collaxa BPEL Server (Collax), can be used to execute the validation processes defined using BPEL4WS. In more detail, as a validation process written in BPEL4WS is inputted into a Collaxa server, the Collaxa server has the built-in ability to (1) test validation process by examining the state of BPEL process instances, (2) track execution and capture the history of the validation process, and (3) monitor the validation process by aggregating statistical information.

It should be noted that the responsibility for monitoring the application of trustworthiness control policies and procedures is different from the overall responsibility for trustworthiness control. Therefore, whenever possible, it is desirable that the two responsibilities be assigned to different roles and individuals.

Monitoring and tracking can also reveal deficiencies of trustworthiness control policies and procedures. Thus, further investigations or corrective actions can be performed, based upon the execution of the validation processes.

The rationale of our proposed framework can be summarized as follows: (1) it is a componentized approach, where each model is built upon an organization foundation and ad hoc Web Services standards; (2) this framework can be adapted and extended to suit the needs of adopting trustworthiness requirements; (3) seamlessly incorporating the most recent standards and typical tools, our framework provides a practical guidance of establishing trustworthiness assurance measurement; and (4) since the language or tool associated with each layer can be replaced by other products without jeopardizing the concept of our framework, our framework will neither impinge upon software vendors' flexibility nor thwart enterprise autonomy.

As a proof of the concept, we have implemented a prototype of the framework (Zhang et al., 2004). However, it should be noted that our framework provides a high-level guidance of establishing trustworthiness

control. Each model for a specific application system needs to be created manually. The quality of the model to be built is fully dependent on the experience of the practitioners. In order to make our framework more practical, we need an integrated development environment tailored to the framework.

ABOUT THIS ISSUE

This issue of the *International Journal of Web Services Research* (JWSR) collects five papers that span from Web Services discovery and development, state management for Web Services composition, information service for Grid computing, and Web service-based personalized Web mining. Special thanks to guest editors Dr. Savas Parastatidis and Dr. Jim Webber for their help in organizing quality papers.

Swapna Oundhakar, et al. address the problem of Web service registration and discovery in a registry federation, which is a collection of autonomous but cooperating Web service registries. They present an ontology-based Web service discovery infrastructure (METEOR-S Web Service Discovery Infrastructure). The discovery algorithm is based upon quantitative measures of the syntactic similarity and the functional similarity between a specified search template and a set of registered Web Services. The empirical evaludation uses a set of 24 Web Services from the stock domain, and preliminary results are reported.

Bing Li and Wei-Tek Tsai propose an ontology-based service-oriented methodology to develop and integrate distributed applications. In their approach, requirements specification are elicited and analyzed based upon a service's point of view, each service is then modeled and described using ontology. That is why the design process is called Ontology and Service Oriented (OSO) programming, and the output of the procedure is called OSO code. Since business logic in OSO code is represented in a machine-understandable format, the subsequent procedure of business process integration can be performed automatically.

Wei Jie, et al. present a hierarchical information service for a computational Grid virtual organization in order to ensure the provision of essential information for a computational Grid. Three layers are identified to support the information service; namely, a virtual organization layer, a site layer, and a resource layer. Based upon performance evaluation of a set of experiments over different models of information data organization, they introduce a novel data organization model. The implementation of their information service is based on the Globus Toolkit and complies with the OGSI (Open Grid Services Infrastructure) specifications.

Marty Humphrey and Glenn Wasson argue that Web Services Resource Framework (WSRF) and WS-Notification are core elements to manage states between Web Services components in order to support effective construction of complex Grid applications. They present an empirical study paper discussing the architectural foundations of WSRF.NET, which is an implementation of the full set of specifications for WSRF and WS-Notification on the Microsoft .NET framework. Their study discusses the architectural implications of the WSRF on the designs and implementations of both WSRF implementations and applications. The observations and lessons learned from the WSRF.NET project provide a basis for further evaluation of the WSRF approach.

Finally, Abdelsalam (Sumi) Helal and Jingting Lu propose a Web service-based information fusion framework that intends to enable end users to collect scattered information from diverse autonomous Web Services. Based upon personal data accumulated, a repeatable process is created transparently by which newer instances of the same information can be obtained in the fu-

ture. A servlet server provides an intermediary broker layer to interact with services.

CONCLUSIONS

The rapidly emerging paradigm of Web Services is widely considered to be the model of the next generation of Internet computing, as it is bridging the gap between business process and IT technology. As numerous Web Services are published on the Internet every day, and as more and more software producers announce their Web service-enabled products (Ferris & Farrell, 2003), there is a potential trustworthiness time bomb, however, lurking in the increasingly popular use of Web Services. Unfortunately, the need to ensure trustworthiness in loosely coupled Web Services that need to be integrated in a seamless fashion requires methodologies that are beyond the current state of the art in this field.

This paper delivers the message of an early recognition of the importance of trustworthy Web Services and envisions a novel layer called WS-Trustworthy to ensure trustworthy Web Services. As the authors have performed some preliminary work, this paper intends to call for further discussions and contributions on this new framework in JWSR.

REFERENCES

BPEL4WS. (2003). *Specification: Business process execution language for Web Services Version 1.1*. Retrieved from *http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/*

Collax. (n.d.). *http://www.collaxa.com*

Ferris, C., & Farrell, J. (2003, June). What are Web Services? *Communications of the ACM, 46*(6), 31.

Mundie, C., Vries, P.V., Haynes, P., & Corwine, M. (2004). *Trustworthy computing* (Microsoft white paper, revised). Retrieved from *http://www.microsoft.com/mscorp/innovation/twc/twc_whitepaper.asp*

Neumann, P. (2004). *Principled assuredly trustworthy composable architectures, emerging draft of the final report for DARPA's composable high-assurance trustworthy systems (CHATS) program, 2004*. Retrieved from *http://www.csl.sri.com/users/neumann/chats4.pdf*

Parnas, D.L., Schouwen, A.J.V., & Kwan, S.P. (1990, June). Evaluation of safety-critical software. *Communications of the ACM, 33*(6), 636-648.

WS-Resources. (n.d.). *http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf*

WS-Security. (2004). *http://www-106.ibm.com/developerworks/webservices/library/ws-secure*

Zhang, J., Zhang, L.-J., & Chung, J.-Y. (2004). WS-trustworthy: A framework for Web Services centered trustworthy computing. *Proceedings of the IEEE International Conference on Services Computing (SCC 2004)*, Shanghai, China, September 15-18, (pp. 186-193).

ENDNOTES

[1] It should be noted that, although computing generally has a broader scope that includes other elements (e.g., hardware, system) in addition to software, we focus here only on software computing, since it is the ultimate deliverable. In this paper, we will use the terms trustworthy computing and software trustworthiness interchangeably.

Jia Zhang (jiazhang@cs.niu.edu) is currently assistant professor of the Department of Computer Science at Northern Illinois University and a guest scientist of the National Institute of Standards and Technology (NIST). Her current research interests center on software trustworthiness in the domain of Web Services, with a focus on reliability, integrity, security, and interoperability. Zhang has published about 50 technical papers in journals, book chapters, and conference proceedings. She also has seven years of industrial experience as a software technical lead in Web application development. Zhang received a PhD in computer science from the University of Illinois at Chicago in 2000. She is a member of the IEEE and ACM.

Liang-Jie Zhang (zhanglj@us.ibm.com) is a chief architect of industrial standards of IBM Software, the founding chair of the Services Computing Professional Interest Community (PIC), and a research staff member at the IBM T.J. Watson Research Center. He is part of the Business Informatics research team with a focus on SOA and Web Services for industry solutions and business performance management services. He has filed more than 30 patent applications in the areas of e-commerce, Web Services, rich media, data management, and information appliances, and he has published more than 80 technical papers in journals, book chapters and conference proceedings. Zhang is the chair of the IEEE Computer Society's Technical Committee on Services Computing and the editor-in-chief of the International Journal of Web Services Research (JWSR). *In 2005 he will serve as the general co-chair of the IEEE International Conference on Web Services (ICWS 2005) and the general co-chair of the IEEE Conference on Services Computing (SCC2005). Zhang received a BS in electrical engineering at Xidian University in 1990, an MS in electrical engineering at Xi'an Jiaotong University in 1992, and a PhD in computer engineering at Tsinghua University in 1996.*