

# Position Paper: Formal Methods for Developing Adaptable, Secure, Situation-aware Service-oriented ( $AS^3$ ) Architectures

Ramesh Bharadwaj<sup>1</sup> Supratik Mukhopadhyay<sup>2</sup>

## Abstract

*In this position paper, we are concerned with formal methods for developing agent-based, situation-aware, secure, survivable architectures for on-demand discovery and composition of web services. In particular, we describe the development of an adaptable, situation-aware, secure architecture that provides for proofs that guarantee the consistency of service level agreements between service providers and their clients. The architecture is based on decentralized situation-aware ambients (SAAs) which are autonomous agents that process information about the current situation, and react to it by discovering and composing services syntactically to adapt to changing situations. The agents run on the Secure Infrastructure for Networked Systems (SINS) under development at the Naval Research Laboratory.*

## I. Introduction of the Authors

*Dr Ramesh Bharadwaj* is a computer scientist at the Naval Research Laboratory where he conducts research on high assurance computer systems. In particular, Dr Bharadwaj was responsible for the development of the synchronous programming language SOL (Secure Operations Language) that allows one to model agents as communicating state machines that can be formally verified. The models can then be used to generate executable code. Prior to joining the Naval Research Laboratories, Dr Bharadwaj has held research positions at the Stanford University and the Bell Laboratories. Dr Bharadwaj performed his doctoral research at the McMaster University under the guidance of Prof. David Parnas.

*Dr Supratik Mukhopadhyay* is an assistant professor at the West Virginia University where he conducts research in the area of formal methods and distributed systems. In particular, Dr Mukhopadhyay has been working on developing programming models for service-based systems. Prior to

joining West Virginia University, Dr Mukhopadhyay was a postdoctoral researcher at the University of Pennsylvania. Dr Mukhopadhyay did his doctoral research on formal verification of embedded software at the Max Planck Institute.

## II. Description of Our Research

This section describes relevant research conducted by the authors sponsored by the Office of Naval Research (ONR)<sup>1</sup>.

### A. Background

On demand *service-oriented architectures* (SOA) are becoming ubiquitous. In an SOA, client applications are developed by composing services, which are platform independent heterogeneous components running on spatially distributed individual nodes of a network. Requests from clients are handled by on-demand discovery and composition of services that satisfy the client's service requirements. Delivery of services to clients is bound by *service level agreements* (SLAs) between the vendors and the clients that additionally specify the *quality of service* (QoS) that the service provider needs to guarantee and the appropriate penalties for their violation. QoS constraints that a service provider guarantees may include security, timeliness, and availability. It is difficult to provide such guarantees when services are spatially distributed over a global network subject to intrusion, congestion and delays.

An SOA should be situation-aware in the sense that in the event of a changing situation, e.g., a service changing its configuration, it should be able to react by providing a revised service composition that continues to meet the client's requirements. To address the security concerns of the client application developers, security and survivability should be provided automatically as an integral part of the architecture. It has been generally accepted that intelligent software agents provide the right building blocks for developing secure and dependable distributed systems. Apart from being fault-tolerant, an agent-based architecture can provide for

<sup>1</sup> Naval Research Laboratory

<sup>2</sup> West Virginia University

<sup>1</sup> The views expressed in this paper are those of the authors and do not in any way reflect the views of the Office of Naval Research or the United States Government

in the sense that individual components can be formally verified for their functionalities and then composed using sound composition rules.

## B. Our Proposed Architecture

Using techniques from formal methods, we have developed an architecture [1], [2], [3], [4] based on decentralized situation-aware ambients (SAA) which are autonomous agents that process information about the current situation, and discover and compose services in order to adapt to the changing situation. These agents run on the Secure Infrastructure for Networked Systems (SINS) [1], a formally verified run-time infrastructure for agents developed at the Naval Research Laboratory. Agents are deployed on SINS Virtual Machines (SVMs) which are responsible for managing agents on the respective hosts.

We have developed a framework for developing semantically correct SAA's using syntactic service composition techniques. The services and resources are specified declaratively at a high level in a modal logic (called  $AS^3$  logic). The modal logic provides atomic formulas that include user-defined atoms, arithmetic constraints for describing real time deadlines and has modalities for describing both time and space constraints. Functional requirements including QoS constraints such as access control and security are specified by the client in a language like WSDL which is eventually compiled into a declarative logical description (in the modal logic). A deductive engine implementing a proof system for the logic is used to generate the model, which comprises a set of SAAs carrying the service composition. An SAA model may be formally verified for application-independent properties like missed deadlines, safety properties (e.g., absence of deadlock), using model checking [5].

## III. Related Work

Existing languages like BPEL4WS [6] for describing business processes that integrate web services lack formal semantics. Heterogeneous plug-ins are needed for describing non-functional requirements like security. Ontology-based languages like OWL-S [7] can provide limited formal guarantees. Existing formal techniques for service composition like SWORD [8] or process calculi-based approaches [9] cannot deal with situation-awareness.

## IV. Addressed Questions

We address the following questions in this position paper

ing business processes at the right level of abstraction that has formal semantics and can express non-functional properties like fault-tolerance, security and timeliness? Can we program business processes in the model whose behavior can vary with situation? In particular, this language can model business processes the same way as the  $\pi$ -calculus [10] models mobile processes.

- Can we define logics for specifying properties of business processes that should not only be able to describe functional properties of the processes but also the QoS that they provide to the clients? Can we find decidable fragments of such logics that can still express interesting properties of business processes? Can we automatically synthesize models of business processes from declarative logical specifications?
- Can we develop suitable static analysis techniques to verify business processes for application-independent properties like deadlock, mutual-exclusion etc.?
- Can we develop run-time monitors that can dynamically monitor business processes for any possible breach of trust and issue corresponding recovery actions? Can we develop a taxonomy for such recovery actions?

## References

- [1] R. Bharadwaj, "Development of dependable component-based applications," in *In Proceedings of the First International Symposium on Leveraging Applications of Formal Methods (ISOLA)*. 2004, LNCS, Springer.
- [2] S. Yau, S. Mukhopadhyay, and R. Bharadwaj, "Specification, analysis, and implementation of architectural patterns for dependable software systems," in *In Proceedings of IEEE International Workshop on Object-oriented, Real-time and Dependable Systems (WORDS)*. 2005, IEEE Computer Society.
- [3] R. Bharadwaj, S. Mukhopadhyay, and N. Padh, "Service composition in a secure agent-oriented architecture," in *In Proceedings of the IEEE International Conference on E-Technologies, E-Commerce and E-Services (EEE)*. 2005, IEEE Computer Society.
- [4] S. S. Yau, H. Davulcu, S. Mukhopadhyay, D. Huang, and Y. Yao, "Adaptable, situation-aware, secure, service-based ( $as^3$ ) systems," in *In Proceedings of the IEEE International Symposium on Object-oriented, Real-time, Distributed Computing (ISORC)*. 2005, IEEE Computer Society.
- [5] E. M. Clarke, O. Grumberg, and D. Peled, *Model Checking*, MIT Press, 2000.
- [6] "Business process execution language for web-services," <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>, 2005.
- [7] "Owl-s," <http://www.daml.org/services/owl-s/1.0/>, 2003.
- [8] S. Ponnekanti and A. Fox, "Sword: A developer toolkit for web service composition," in *In Proceedings of the World Wide Web Conference (WWW)*. 2002, <http://www2002.org/CDROM/alternate/786/>.
- [9] S. J. Woodman, D. J. Palmer, S. K. Shrivastava, and S. M. Wheeler, "Notations for the specification and verification of composite web services," in *In Proceedings of the International Enterprise Distributed Object Computing Conference (EDOC)*. 2004, IEEE Computer Society.
- [10] A. J. R. J. Milner, *Communicating and Mobile Systems: the  $\pi$ -Calculus*, Cambridge University Press, 1999.