

Service Choreography and Orchestration with Conversations

Tevfik Bultan

Department of Computer Science
University of California, Santa Barbara
bultan@cs.ucsb.edu

Service oriented computing provides technologies that enable multiple organizations to integrate their businesses over the Internet. Typical execution behavior in this type of distributed systems involves a set of autonomous peers interacting with each other through messages. Modeling and analysis of interactions among the peers is a crucial problem in this domain due to following reasons: 1) Organizations may not want to share the internal details of the services they provide to other organizations. In order to achieve decoupling among different peers, it is necessary to specify the interactions among different services without referring to the details of their local implementations. 2) Modeling and analyzing the global behavior of this type of distributed systems is particularly challenging since no single party has access to the internal states of all the participating peers. Desired behaviors have to be specified as constraints on the interactions among different peers since the interactions are the only observable global behavior. Moreover, for this type of distributed systems, it might be worthwhile to specify the interactions among different peers before the services are implemented. Such a top-down design strategy may help different organizations to better coordinate their development efforts.

This type of distributed systems can be modeled as a composite Web service that consists of a set of peers that interact with each other via synchronous and/or asynchronous messages [3]. A conversation is the global sequence of messages exchanged among the peers participating to a composite Web service [2]. A choreography specification identifies the set of allowable conversations for a composite Web service. An orchestration, on the other hand, is an executable specification that identifies the steps of execution for the peers.

This conversation based modeling framework leads to the following interesting problems: realizability, synthesis, conformance, and synchronizability [4]. A choreography specification is realizable if the corresponding conversation set can be generated by a set of peers [6, 9]. This step is necessary to guarantee that the choreography specifications that are developed in a top-down manner are implementable. A related problem is automated synthesis of peer implementations from a given choreography specification. Another interesting problem is investigating the conformance between orchestration and choreography specifications. An orchestration specification conforms to a choreography specification if the global sequence of messages generated by the orchestration is allowed by the choreography specification. Finally, synchronizability analysis [5, 10] investigates the effects of asynchronous vs. synchronous communication to improve the

efficiency of interaction analysis. A set of asynchronously communicating peers are synchronizable if their conversation set does not change when asynchronous communication is replaced with synchronous communication. Replacing asynchronous communication with synchronous communication enables more efficient analysis by removing the communication channels from the state space of the system.

Web Service Analysis Tool (WSAT) [7] is a tool for analyzing conversations. WSAT verifies LTL properties of conversations, checks sufficient conditions for realizability and synchronizability and synthesizes peer implementations from choreography specifications. In order to model XML data, WSAT uses a guarded automata model where the guards of the transitions are written as XPath expressions. WSAT uses the explicit-state model checker SPIN [11] for LTL model checking by translating the guarded automata model to Promela [8]. WSAT has been used to analyze realizability and synchronizability of composite Web services specified using BPEL orchestration language [1] and conversation protocols [6], which is a formalism for choreography specification and analysis.

References

1. Web services business process execution language version 2.0. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>, April 2007.
2. T. Bultan, X. Fu, R. Hull, and J. Su. Conversation specification: A new approach to design and analysis of e-service composition. In *Proceedings of the Twelfth International World Wide Web Conference*, pages 403–410, May 2003.
3. T. Bultan, X. Fu, and J. Su. Analyzing conversations of web services. *IEEE Internet Computing*, 10(1):18–25, 2006.
4. T. Bultan, X. Fu, and J. Su. Analyzing conversations: Realizability, synchronizability, and verification. In Luciano Baresi and Elisabetta Di Nitto, editors, *Test and Analysis of Web Services*, pages 57–85. Springer, 2007.
5. X. Fu, T. Bultan, and J. Su. Analysis of interacting BPEL web services. In *Proceedings of the 13th International World Wide Web Conference*, pages 621–630, May 2004.
6. X. Fu, T. Bultan, and J. Su. Conversation protocols: A formalism for specification and verification of reactive electronic services. *Theoretical Computer Science*, 328(1–2):19–37, November 2004.
7. X. Fu, T. Bultan, and J. Su. WSAT: A tool for formal analysis of web services. In R. Alur and D. Peled, editors, *Proceedings of the 16th International Conference on Computer Aided Verification*, volume 3114 of *Lecture Notes in Computer Science*, pages 510–514. Springer-Verlag, July 2004.
8. X. Fu, T. Bultan, and J. Su. Model checking XML manipulating software. In *Proceedings of the ACM/SIGSOFT International Symposium on Software Testing and Analysis*, pages 252–262, July 2004.
9. X. Fu, T. Bultan, and J. Su. Realizability of conversation protocols with message contents. *International Journal of Web Services Research*, 2:68–93, 2005.
10. X. Fu, T. Bultan, and J. Su. Synchronizability of conversations among web services. *IEEE Trans. Software Eng.*, 31(12):1042–1055, 2005.
11. G. J. Holzmann. The model checker SPIN. *IEEE Transactions on Software Eng.*, 23(5):279–295, May 1997.