Realizability of Conversation Protocols with Message Contents

Xiang Fu, Georgia Southwestern State University, USA
Tevfik Bultan, University of California, Santa Barbara, USA
Jianwen Su, University of California, Santa Barbara, USA

ABSTRACT

A promising way to model the global behavior of a Web Service composition is to characterize the set of conversations among the participating Web Services. A conversation protocol specifies the desired global behaviors of a Web Service composition. The realizability problem is to decide whether, given a conversation protocol, a Web Service composition can be synthesized that generates exactly the same set of conversations specified by the protocol. This is a key problem in the top-down specification of the Web Service compositions. In our earlier work, we developed sufficient conditions for realizability of conversation protocols based on a model that abstracts the contents of the messages. The present paper extends our earlier work by allowing message contents to be used in the realizability analysis. We show that taking the message contents into account yields more accurate analysis. To overcome the state-space explosion caused by the message contents, we propose symbolic analysis techniques for the realizability conditions. In addition, we show that the analysis of one of the realizability conditions — the autonomy condition — can be done using an iterative refinement approach.

Keywords: asynchronous communication; conversation protocols; messages; realizability; verification; Web Service composition

INTRODUCTION

Constructing highly dependable Web Service compositions is a challenging task because of their distributed nature. At the same time, without ensuring the dependability of the Web Service compositions, it would not be advisable to use the Web Service framework in critical applications. Recently, automated verification of Web Service compositions has attracted significant attention in order to address this problem (e.g., Foster, Uchitel, Magee, & Kramer 2003; Fu, Bultan, & Su, 2004a; and Narayanan & McIlraith, 2002). However, application of the automated verification techniques to Web Services is not trivial. One challenge is the establishment of a formal modeling and specification framework for Web Service compositions. There are numerous competing standards for composition of Web Services (e.g.,
BPEL4WS [BPEL, n.d.], WSCI [WSCI, n.d.], OWL-S [DAML-S, n.d.], which complicates this task. Additionally, certain characteristics of Web Services cause important problems. For example, asynchronous communication (supported by messaging platforms such as Java Message service [JMS, n.d.] and Microsoft Message Queuing service [MSMQ, n.d.]) may significantly increase the complexity of many verification problems and, in some cases, make the problems undecidable (Fu, Bultan, & Su, 2003). An emerging paradigm of Web Service composition is to use conversations to describe interactions among participating Web Services (Hanson, Nandi, & Kumaran, 2002; Hanson, Nandi, & Levine, 2002; Banerji et al., 2002; Bultan, Fu, Hull, & Su, 2003; Fu, Bultan, & Su, 2003). A core idea common in these models is the use of finite state machines to represent some aspects of the global composition process. The state machines can involve two parties (Hanson, Nandi, & Kumaran, 2002; Hanson, Nandi, & Levine, 2002) or multi-parties (Bultan, Fu, Hull, & Su, 2003; Fu, Bultan, & Su, 2003) and may describe the global composition process directly (Hanson, Nandi, & Kumaran, 2002; Hanson, Nandi, & Levine, 2002; Bultan, Fu, Hull, & Su, 2003; Fu, Bultan, & Su, 2003) or may specify its local views (Banerji et al., 2002; Bultan, Fu, Hull, & Su, 2003; Fu, Bultan, & Su, 2003). In our previous work reported by Bultan, Fu, Hull, and Su (2003), we established a conversation-oriented framework to specify Web Service compositions and to reason about their global behaviors. Each participant (peer) of a composition is characterized using a finite state automaton (FSA) with the set of input/output message classes (without message contents) as the FSA alphabet. To capture asynchronous communication, each peer is equipped with a FIFO queue to store incoming messages. The behaviors generated by a composition of peers can be characterized using the set of message sequences (conversations) exchanged among peers. Linear Temporal Logic (LTL) naturally is extended to this conversation-based framework (Fu, Bultan, & Su, 2003) and can be used to specify desired system properties, such as “when a cancel request arrives, eventually the server should respond with a cancel confirmation message.”

In the general context of our framework, there are two different ways to specify a Web Service composition:

1. The bottom-up approach (favored by many industry standards, including WSDL [n.d.] and BPEL4WS [BPEL, n.d.], in which each participant of the composition is specified first and then the composed system is studied; and

2. The top-down approach (e.g., conversation policies [Hanson, Nandi, & Kumaran, 2002] and Message Sequence Charts [MSC, 1994; Foster, Uchitel, Magee, & Kramer 2003]), in which the set of desired message sequences is specified, and detailed peer implementations are left blank.

Bultan, Fu, Hull, and Su (2003), generalized the notion of a conversation policy from two peers to an arbitrary number of peers and proposed the notion of a conversation protocol. Although the expressive power of a conversation protocol is weaker than that of the bottom-up approach, the top-down approach provides an advantage in that verification of composition properties can better utilize existing algorithms and tools developed in the verification community (Bultan, Fu, Hull, & Su, 2003). A conversation protocol is not always realizable in the sense that there may not exist any peer implementations whose composition generates exactly the same set of conversations as specified by the protocol. Fu, Bultan, and Su (2003), proposed sufficient conditions for realizability. The three realizability conditions proposed restrict control flows of a conversation protocol, and when these three conditions are satisfied, the projections of the protocol to each peer are guaranteed to be a realization of the protocol. One advantage of the realizability analysis is that it avoids the undecidability of