



Toward the Right Communication Protocol for Web Services

Zahir Tari, RMIT University, Melbourne, Australia
Manish Malhotra, RMIT University, Melbourne, Australia
Abdelkamel Tari, University of Bejaia, Berber State, Algeria
Mark McKinlay, RMIT University, Melbourne, Australia

ABSTRACT

Web Services require a common understanding of messages and message content in order to interact with each other. Several protocols have been proposed, which are coded into agent interfaces or implemented using framework-specific methods such as shared repositories. However, an agent wishing to interact with a large number of other disparate agents cannot be expected to have knowledge of how to initiate or maintain a conversation with all other agents it may encounter. In this paper, we propose a dynamic communication protocol for Web services—DynWES. This involves the publication of protocol specifications representing a finite state machine (FSM). A client agent downloads this specification, validates it for correctness, and then implements the protocol dynamically as a state machine. Mechanisms for building FSMs and checking their correctness are presented. An implementation of DynWES also is described and tested (using applications in the wine selling business domain).

Keywords: agents; communication protocol; conversation protocol; protocol correctness; state machines; Web Services

BACKGROUND

Web service platforms provide the required functionalities to enable large-scale integration of business applications across enterprises (Carey et al., 2001). They provide solutions for a business domain where services or their agents need to interact to provide the required solution. This may involve a simple client/server relationship between a Web server and a client application, or an enterprise solution composed of distributed services. Communication

between entities usually requires a reliable transport of the communication, a common understanding of the data being exchanged, and an understanding of the sequence of exchanges, forming a valid communication protocol.

Of these requirements, defining communication protocols may be the most problematic. While Web services may use well-known transport layer protocols and implementation languages, communication protocols are application dependent. Clients wishing to make use of a service must

know the message formats and valid sequence of message exchange that forms the conversation expected by the service provider. This implies a prior knowledge of the conversation requirements by the client. As the number of Web services increases in a large environment such as the Internet, agents increasingly may be required to work through the large number of services and actions available. Under such circumstances, conversations may need to be discovered dynamically rather than via prior knowledge, and this evolution may be based on open technologies (Ma, 1999), such as CORBA² (supports an object-oriented remote procedural call) and SOAP³ (an XML-based remote procedural call). Of course, there also are other technologies such as Microsoft's .Net.

Several works have been published in the area of software agents. A well-known example is ACL (Agent Communication Language) proposed by Finin et al. (1994). Typically, their communication protocols often are not standardized and may offer little interoperability. This often leads to proprietary interfaces and protocols, but this may still provide poor interoperability between different types of agents. While frameworks exist to improve agent interoperability (e.g., IBM Aglets workbench⁴, Voyager⁵, Java Net Agents⁶), agents operating across the Internet cannot be assumed to work in any such framework and must try to operate with whatever accepted or de facto standards exist for the Internet. For an agent operating in the Internet, its execution environment is not a single controlled framework but rather a large heterogeneous environment where all expected conversations could not be anticipated. For example, a simple shopping agent may wish to interact with multiple merchant sites to retrieve product information and purchase goods, where each

site has been constructed with its own communication protocols and data requirements.

Reaching a consensus about standards in an environment of heterogeneity and proprietary solutions is highly problematic. Therefore, interoperation between information agents in the Internet may be viewed as the goal (Genesereth & Ketchpel, 1994) rather than an implementation of any standard. This might be achieved by using de facto standards (Chowdhry & Hughes, 1997) (e.g., XML, HTTP, SOAP) and negotiation of protocols. In these conditions, one agent would not care what standard(s) another agent implemented, provided there was understanding between the agents about what communication was required (e.g., "I do not care what your standard is, just tell me what to speak and I will speak it.").

While several solutions using XML/SOAP and HTTP have been proposed (i.e., xCBL⁷ and SCMP) (Arnold et al., 2001), these do not define how any transaction is performed between agents. For example, a merchant site may support a certain format of data messages, but there still remains the issue of how these messages can be used to form a valid sequence comprising a conversation protocol. Merchant sites may use the same message formats but may use them in a different sequence. Another site might use different messages altogether to provide the same service.

To provide full interoperability between information agents on the Internet, we need agents not only to know the correct data formats to pass (e.g., SOAP syntax), but also to know the conversation level protocol involving those messages for any required service. This also must include the valid responses, where multiple responses are possible, and the starting and ending states of the conversation. In this paper,

22 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:

www.igi-global.com/article/toward-right-communication-protocol-web/3058?camid=4v1

This title is available in InfoSci-Journals, InfoSci-Journal Disciplines Computer Science, Security, and Information Technology, InfoSci-Digital Marketing, E-Business, and E-Services eJournal Collection, InfoSci-Networking, Mobile Applications, and Web Technologies eJournal Collection, InfoSci-Journal Disciplines Business, Administration, and Management, InfoSci-Select. Recommend this product to your librarian:

www.igi-global.com/e-resources/library-recommendation/?id=2

Related Content

Adaptive Ensemble Multi-Agent Based Intrusion Detection Model

Tarek Helmy (2010). *Developing Advanced Web Services through P2P Computing and Autonomous Agents: Trends and Innovations* (pp. 36-48).

www.igi-global.com/chapter/adaptive-ensemble-multi-agent-based/43646?camid=4v1a

Modeling and Respecting Privacy Specification when Composing DaaS Services*

Salah-Eddine Tbahrity, Brahim Medjahed and Chirine Ghedira (2012). *International Journal of Web Services Research* (pp. 24-44).

www.igi-global.com/article/modeling-and-respecting-privacy-specification-when-composing-daas-services/80177?camid=4v1a

A Service-Oriented Composition Framework with QoS Management

Casey K. Fung, Patrick C.K. Hung, Richard C. Linger, Guijun Wang and Gwendolyn H. Walton (2006). *International Journal of Web Services Research* (pp. 108-132).

www.igi-global.com/article/service-oriented-composition-framework-qos/3086?camid=4v1a

Dynamic, Flow Control-Based Information Management for Web Services

Zahir Tari, Peter Bertok and Dusan Simic (2008). *Web Services Research and Practices* (pp. 260-303).

www.igi-global.com/chapter/dynamic-flow-control-based-information/31216?camid=4v1a