

A Dependable Infrastructure for Cooperative Web Services Coordination

Eduardo Adilio Pelinson Alchieri, Federal University of Santa Catarina, Brazil

Alysson Neves Bessani, University of Lisbon, Portugal

Joni da Silva Fraga, Federal University of Santa Catarina, Brazil

ABSTRACT

A current trend in the web services community is to define coordination mechanisms to execute collaborative tasks involving multiple organizations. Following this tendency, in this paper the authors present a dependable (i.e., intrusion-tolerant) infrastructure for cooperative web services coordination that is based on the tuple space coordination model. This infrastructure provides decoupled communication and implements several security mechanisms that allow dependable coordination even in presence of malicious components. This work also investigates the costs related to the use of this infrastructure and possible web service applications that can benefit from it.

Keywords: Coordination Mechanisms, Dependability, Tuple Spaces, Web Service Applications, Web Services Coordination

INTRODUCTION

The Web Services technology, an instantiation of the service oriented computing paradigm (Bichier & Lin, 2006), is becoming a *de facto* standard for the development of distributed systems on the Internet. The attractiveness of web services are its interoperability and simplicity, based on technologies widely used in the web, like HTTP and XML. Its attributes have been the motivation for many industrial and academic efforts for developing concepts

and models for distributed applications based on the service-oriented paradigm.

The service oriented computing - and more specifically Web Services - is a natural evolution of classical concepts such as RPC and technologies like CORBA (Object Management Group, 2002). In the same way of these, web services provided by an organization must be described in an interface that can be understood and invoked by other parts of a distributed system. The main standards for web services, all based on XML, are: SOAP (*Simple Object Access Protocol*) - protocol for exchange messages among clients and services, which can

DOI: 10.4018/jwsr.2010040103

operate over several communication protocols; the WSDL (*Web Service Description Language*) - language used to describe web services; and the UDDI (*Universal Description, Discovery and Integration*) - the repository where the web services are registered in order to be discovered by clients.

The main goal of web services is the interoperability. Thus, many efforts have appeared to define adequate forms of services composition in order to execute complex tasks. These compositions aim the cooperation in the execution of tasks involving multiple organizations (Bichier & Lin, 2006; Peltz, 2003). Specifications like *WS-Orchestration* (Alves, 2006) and *WS-Choreography* (Burdett & Kavantzias, 2004) address exactly this problem, specifying mechanisms for definition and execution of tasks that involve several web services.

Current approaches, as the cited above, aim the web services integration through the specification of message flows among the services (control-driven coordination (Papadopolous & Arbab, 1998)). Another approach, which complements the first one, is the web services coordination through the use of a shared data repository (data-driven coordination (Papadopolous & Arbab, 1998)). In this approach, the cooperating services communication is done through a shared data repository, which can be used as data storage or as a coordination mediator, providing decoupled communication. *Tuple spaces* (Gelernter, 1985) are a popular data-driven coordination model.

In the tuple space model, processes interact through a shared memory abstraction, the tuple space, where generic data structures called tuples are stored and retrieved. The basic operations supported by the tuple space are insertion, reading and removal of tuples. The main attractiveness of this model is its support for communication that is decoupled both in time (communicating processes do not need to be active at the same time) and space (communicating processes do not need to know each others locations or addresses) (Cabri et al., 2000).

There has been some research about the integration of web services using the tuple space model (Bright & Quirchmayr, 2004; Lucchi & Zavattaro, 2004; Maamar et al., 2005; Bellur & Bondre, 2006). The main advantage of this approach is the decoupling among the cooperating services: not even the services interfaces needs to be completely known. This work follows in the same line and proposes a cooperation infrastructure for web services that provides the inherent benefits of the tuple spaces model but that is also dependable.

The infrastructure proposed in this paper, called **WS-DependableSpace** (WSDS), extends previous works of the authors on tuple space dependability (Bessani et al., 2008; Bessani et al., 2009), incorporating new components that provide the integration of a dependable tuple space on the web services world. The WSDS architecture relies on stateless gateways that execute operations on a dependable tuple space, forwarding client requests from web service clients to it and vice-versa. Several mechanisms have been introduced in this architecture in order to tolerate accidental faults (like crashes and software bugs) as well as malicious attempts to disable the operation of system components (like attacks and intrusions) (Verissimo et al., 2003). Moreover, WSDS maintains all the dependability properties of a dependable tuple space (Bessani et al., 2008) *without further extensions to the web services core specifications*.

Why use WSDS to Coordinate Web Services?

Since or objective with WSDS is to provide a coordination infrastructure for web services, the attentive reader can ask: Why someone would use a service like WSDS to coordinate web services, instead of using some tool that implements web services coordination specifications? Here we list several reasons to use a service like WSDS to coordinate web services.

In usual web services coordination, the communication is often coupled in at least two levels: (1) cooperative services must be

20 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/dependable-infrastructure-cooperative-web-services/42109?camid=4v1

This title is available in InfoSci-Journals, InfoSci-Journal Disciplines Computer Science, Security, and Information Technology. Recommend this product to your librarian:

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

Related Content

Redundancy-Based Reliability Enhancement for Composite Services

Zhongjie Wang, Shaopeng Wang, Dianhui Chu and Xiaofei Xu (2014). *International Journal of Web Services Research* (pp. 27-51).

www.igi-global.com/article/redundancy-based-reliability-enhancement-for-composite-services/122814?camid=4v1a

F-DRARE: A Framework for Deterministic Runtime Adaptation of Cyber Physical Systems

Fahad Bin Tariq and Sandeep Korrapati (2013). *Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions* (pp. 263-276).

www.igi-global.com/chapter/drare-framework-deterministic-runtime-adaptation/69478?camid=4v1a

RAW Architecture: Reflective and Adaptable Web Service Model

Javier Parra-Fuente, Salvador Sanchez-Alonzo, Oscar Sanjuan-Martinez and Luis Joyanes-Agular (2005). *International Journal of Web Services Research* (pp. 36-53).

www.igi-global.com/article/raw-architecture-reflective-adaptable-web/3054?camid=4v1a

Metadata, Ontologies, and Information Models for Grid PSE Toolkits Based on Web Services

Carmela Comito, Carlo Mastoroiani and Domenico Talia (2006). *International Journal of Web Services Research* (pp. 52-72).

www.igi-global.com/article/metadata-ontologies-information-models-grid/3089?camid=4v1a