Enhancing OSGi: Semantic add-ins for Service Oriented Collaborative Environments

Pablo Cabezas[1,2], Raúl Barrena[1], Jon Legarda[1,2], Diego López de Ipiña[3]

[1]CEIT and [2]Tecnun (University of Navarra)
Manuel de Lardizábal 15, 20018 San Sebastián, Spain
{pcabezas, rbarrena, jlegarda}@ceit.es

[3]Faculty of Engineering (ESIDE), University of Deusto
Avda. Universidades 24, 48007 Bilbao, Spain
dipina@eside.deusto.es

Abstract. Service Oriented Architectures offer an incomparable setting for the management and reuse of services, mixing different factors like software and services. The ability to choose between the services available often gets blurry, because of the difficulty when trying to find the best service that fits better the actual needs, or even because its invocation process gets excessively complex. This work presents a horizontal layer that enhances service capabilities in OSGi by adding semantic information for intelligent and advanced management in SOA-compliant smart architectures, and more specifically in a Multi-Residential Environment.

Keywords: Service Oriented Architectures, Semantic Services, Multi-Residential Environments, OSGi

1 Introduction

Traditionally considered as a heterogeneous 'left-over' collection of activities that are not included in the agriculture or industry sectors, the services/software sector has, until recently, been a neglected area of economic policy making. As companies learn to trade products and services in new ways, often through Information and Communication Technologies (ICTs), services have become a pillar of the European economy [1]. In fact, every sector of the economy is moving (or planning to move) its business to the electronic way [2].

The 'Service' term often looks abstract, covering areas in different and heterogeneous sectors. The Service Oriented Architecture (SOA) approach covers perfectly this idea by creating new solutions centered in the service, the key piece for communications and value search of the mentioned actors in the structure.

But still, those services must be enriched in order to get the most of their potential and create truly scalable and healthy architectures. This work shows the addition of a semantic layer in the top of a Multi-Residential Gateway[3-5] SOA-compliant Architecture, based on the Residential Gateway approach [6]. It creates a horizontal structure for dynamic management and intelligent service provisioning, based on The Semantic OSGi Service Architecture [7-8]. It enriches service declarations, allowing intelligent and automatable searching and invoking processes, and can be applied to different SOA based environments in the actual computational structural model.

The paper is organized as follows: Section 2 presents the terms and challenges in the union of Software, Services and Semantics, showing the application of the Distributed Semantic OSGi service aRchitecture (DSOR) to a Multi-Residential real solution, while in Section 3 the main conclusions and future work are pointed out.

2 DSOR: Smart Services for Intelligent Environments

Services are the main piece in the service oriented puzzle, and the reason for the relationship established between Service Providers and Requesters. Initially solutions were made in an absolute ad-hoc way, providing strict, non-scalable, non-modular distributed applications. Nowadays, this situation is changing as more efficient solutions are being demanded by the Information and Services Society. Within this context, information needs to be processed in a ubiquitous and pervasive way, as well as interconnected by means of smart embedded computing, building up a new broadband ambient intelligent paradigm.

The DSOR solution is presented below, a service management solution that uses semantic technologies for intelligent service management in smart environments, and that has been successfully applied to a Multi-Residential Environment with MRG units.

2.1 Semantic Service Management + OSGi

Among the solutions proposed, the vast majority are based on the Open Standard Gateway Initiative (OSGi) specification [9-10], which promotes the dynamic discovery and collaboration of devices and services from different sources. The OSGi service framework provides a simple lightweight framework for creating service-oriented applications. Based on this framework, the use of technologies like web services is extremely useful and makes it possible to support interoperable machine-to-machine interactions over a network. The physical and logical infrastructure must be complemented with an intelligent service management architecture.

Several service management platforms have been created over the OSGi framework for distributed environments, using semantic technologies to index and catalogue the services that will subsequently be required by the different consumers in the scenario [11-12]. One of these solutions is the Dynamic Semantic OSGi Service Architecture (DSOR). The addition of semantic information to the OSGi registry was a need in order to facilitate the access through terms in shared vocabularies better than through interfaces and rough traditional programming.

DSOR provides an easy way to publish additional information about services, without modifying the default performing of the OSGi framework. DSOR supposes an evolution of the Semantic OSGi Service Registry (SOR), applying this semantic architecture to the scenario of the MRGs. It also complements the actual OSGi service registry, adding functionalities that do not prevent the default search and invocation engine of the OSGi framework. The SOR in-memory registry makes it possible, providing new searching capabilities that go further from local search and work, adding new layers to the OSGi architecture, without the need of any additional library bundle.

2.2 Annotating Services

The information about services in DSOR must be obtained from the services themselves. This solution makes use of the Java Reflection and the Java Annotation APIs to include the information in the source code and to obtain it later, during execution time.

In Fig. 1 an example about the annotation of a service in DSOR is shown.

```
@S3Method(domain="NetworkManagement",
        concept="add_vlan_interface",
        description="Add VLAN interface to a platform",
        arguments={
            @S3MethodArgument(order=1, concept="host_ip",
                description="Host IP address"),
            @S3MethodArgument(order=2, concept="vlan_id",
                description="Virtual LAN identifier"),
            @S3MethodArgument(order=3, concept="vlan_ip",
                description="IP address for the VLAN interface"),
            @S3MethodArgument(order=4, concept="netmask",
                description="Mask for the VLAN interface"),
            @S3MethodArgument(order=5, concept="interface_name",
                description="Interface name for the VLAN")
        }
    )
public boolean addVlanInterface(String hostIp, int vlanId,
    String ip, String mask, String ifaceName) {
{
```

Fig. 1. DSOR service annotation

Using terms in common shared vocabularies, this is the only coupling point required for the server-client sides to be able to cooperate in the architecture, avoiding the extensive use of Interfaces in the common operation mode of the OSGi Framework. Therefore, DSOR does not need to be directly referenced by the different service-providing and/or client bundles in the framework.

3 Conclusions and Future Work

In the same way as human beings establish a communication thank to language, the complexity in relationships at this level can be decreased with the understanding of certain terms in semantic and well defined sets. The system extensibility, by means of a shared use of resources, is one of the goals in Service Oriented Architectures. Reusability is pretty much a done deal with the correct definition of resources available in the cooperative network, being understandable by every involved part.

DSOR has finally got to be a horizontal solution with applications in heterogeneous projects. The addition of semantic information to services in the environment facilitates the client access through terms in shared vocabularies, better than through interfaces and rough traditional programming.

This way, new logic layers and inference engines can work together and be built in the Framework, in order to add as much complexity as needed in each case. This additional abstraction layer also adds many possibilities for scalability and update tasks in the solution, creating dependencies only at the time of creating the basic layers. It creates data dependencies only with a list of terms that can be updated and communicated with another vocabularies, forming the necessary consensus for both client and server parts.

The performance of the solution has also been evaluated through a battery of tests that consider different scenarios, getting execution times that vary from 30 milliseconds (in positive scenarios, in which services can be obtained) to 42 milliseconds (in the worst case, if the service can not be found, with one thousand different services available in the system). This has also helped to apply load-balancing techniques in the different developed architectures.

Regarding to the future work, the development team is, at the present, extending the system to make it compliant with external semantic service-providing source systems using OWL-S.

References

1. Networked European Software & Services Initiative (NESSI): NESSI Strategic Research Agenda: Strategy to build NESSI, vol. 2 (2007)
2. Iniciativa Española de Software y Servicios (INES): INES: Agenda Estratégica de Investigación, v3.0 (2008)
3. P. M. Bull, P. R. Benyon, and P. R. Limb: Residential gateways. Bt Technology Journal, vol. 20, pp. 73-81 (2002)
4. F.T.H den Hartog, M. Balm, C. M. de Jong and J.J.B. Kwaaitaai: Convergence of residential gateway technology. IEEE Communications Magazine 39, 110-114 (2004)
5. T. A. Nguyen and J. Bouwen: The next-generation residential gateway. Journal of The Institution Of British Telecommunications Engineers 2, 134-138 (2001)
6. S. Arrizabalaga, P. Cabezas, J. Legarda, A. Salterain: Multi-Residential Gateway: an Innovative Concept and a Practical Approach. IEEE Transactions on Consumer Electronics (2008)
7. P. Cabezas, S. Arrizabalaga, A. Salterain, and J. Legarda: An Agent Based Semantic OSGi Service Architecture. Studies in Computational Science, Springer (2008)
8. P. Cabezas, S. Arrizabalaga, A. Salterain, and J. Legarda: SOR: Semantic OSGi service Registry. Presented at UCAmI 2007, Zaragoza, Spain, (2007)
9. O. Alliance: OSGi Service Platform Core Specification, 4 ed (2005)
10. O. Alliance: OSGi Service Platform Service Compendium (2006)
11. Rebeca P. Diaz Redondo, Ana Fernández Vilas, Manuel Ramos Cabrer, José Juan Pazos Arias, Jorge García Duque and Alberto Gil Sola: Enhancing Residencial Gateways: A Semantic OSGi Platform. IEEE Intelligent Systems 23, no. 1, 32-40 (2008)
12. Panagiotis Gouvas, Thanasis Bouras and Gregorio Mentzas: An OSGi-Based Semantic Service-Oriented Device Architecture. On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops, Springer, col. 4806/2007, pp. 773-782 (2007)