Smart Objects For Intelligent Applications - ADK

Jesús Fernández Gómez-Pimpollo
Indra
jfgpimpollo@indra.es

Raúl Otaolea
ESI Tecnalia
Raul.Otaolea@esi.es

Abstract

The poster attempts to describe the ADK tools developed in the SOFIA project (http://www.sofia-project.eu/). The SOFIA project is a three-year ARTEMIS project involving twenty two partners from four different EU countries, its key outcomes of the project relate to user interaction paradigms for interacting in smart environments, the common interpretability solution between many heterogeneous devices and embedded systems, and on the application development schemes that can mobilize new developers for smart environments.

1. Introduction

The ADK is a set of tools that provides several functionalities to the whole development life cycle. The Ontology Driven Development (ODD) [1] based on Ontology Driven Software Engineering approach has three phases: design, implementation, testing and deployment.

2. Design

The Wizard plug-in centralizes the programmer decisions about the Knowledge Processor (KP). The wizard guides the developer to choose in between ontology concepts to use and their profile (producers and/or consumers), platform and programming language, communication protocol and project details like name, package names, etc.

A new SOFIA project is created in the last wizard´s step where all necessary middleware (all layers in the proper language, an editor for that language) is added and finally all the connectors selected, SSAP or communication protocol knowledge is not needed to start coding the logic of a KP. The plug-in selects a proper editor for the programming language chosen by the KP developer, and assigns it to the SOFIA project generated.

Connectors are already coded for a variety of languages (partially programmed for Java) and will be stored in a centralized server where the ADK connects to and downloads locally.

Openness and scalability are major factors, therefore connectors for different communication protocols and different languages are feasible. The Smart Application Wizard includes the OWL2Classes facility, which translates several ontologies into programming languages classes in order to be used with the SOFIA middleware. When this class is used, all its interactions are sent to the SIB; therefore programmers do not need to have expertise knowledge about ontologies.

3. Implementation

A KP is divided in four layers: 1. Logic. A developer is only responsible for programming this layer and it represents the functionalities a final user or a company wants to create. 2. Semantic model. This layer is automatically generated by the OWL2Classes tool of the ADK. It is an API with the classes in a specific programming language representing ontologies selected in the wizard tool. Thus, they are the concepts to exchange with the smart space. 3. SSAP. This layer gathers all APIs dedicated to deal with the SIB. The upper layer knows how to invoke the appropriate methods of this API.[3] Connection. The different connectors are plugged in this layer to connect with the SIB. It is also responsible for Semantic Information Broker (SIB) discovery.

Hence, the complexity related with ontologies and communication protocols are totally encapsulated in a very easy API formed by plain classes (model layer). Developers only have to instantiate them and use their getters/setters in order to interact with the smart space.

4. Testing and Deployment

The SIB is the responsible for the interoperability between heterogeneous smart applications and devices. Its core is differentiated in three parts; 1.Sessions. Refers to the KP connected to the SIB at a given moment. 2. Subscriptions. Subscribed nodes are notified when certain conditions in the semantic model are produced. 3. Semantic Model provides the interoperability level needed to connect different devices, platforms, sensors or data using the SOFIA Smart Space context. It also enables the knowledge processing, sharing and reuse between applications.

The SIB OSGi Bundle exposes a set of services using the public interfaces **ISIB** (shows all the methods to interact with the SIB using SSAP Message type communication), which controls the communication between Gateways/KP and SIB, and **IViewer** (exposes methods to observe the content and to register changes within the SIB)

The Semantic Model is based on Jena (Semantic Web Framework), which provides an API to allow the storage of ontologies, classes, individuals, properties and all data related to ontologies. It is assumed that languages can be used for the interaction with the model are OWL (or RDF-XML) and N3. Decisions are based on the definition of the SSAP Message Protocol defined in Description of XML encoding of SSAP document (internal Wiki). This model maintains a full class hierarchy of already connected nodes and the individuals (instances or triples) that has joined the SIB. All KPs must send in their first ontology model connection to the SIB, so that, all individuals defined in that KP can be understood by the SIB and other KPs in the Smart Space. When a KP is removed from the Smart Space all the information about this KP is also removed from the SIB Semantic Model, including all the inserted individuals.

The ADK **Subscription Manager** is responsible for maintaining the information about the subscribed KPs and **Session Manager** manages the KPs connected to the SIB at a given moment. Only connected KP-nodes are allowed to perform the SSAP operations in the SIB. The SIB into a Sofia smart space needs to interoperate with the KPs of several smart applications that are running into the smart space, which enables the KP to change information into the SIB and the SIB communicates those change to several KPs subscribed to them. KP can be executed in any physical device with communication capabilities to interoperate with a SIB of the smart space. The use of Gateways is a solution to avoid problems when having devices with different communication protocols.

A modular and extendible architecture is achieved by having the OSGI platform as a module container, where the SIB or the Gateways are OSGI bundles and SIB bundle changes will not affect any Gateway bundle. This architecture will be enhanced with new Gateways supporting new communication protocols and increasing the amount of devices where smart applications can be deployed, freeing new Gateway developers of any responsibility regarding opened sessions (KPs-SIB) as the fragment bundle will be extended by the Gateway.

Moreover, two specific Gateway OSGI Bundles are been looked at the moment to support **TCP/IP** and **Bluetooth** connections. The platform is also open to develop new Gateways supporting more connection protocols.

SIB includes a server to represent any server visually, which main functions are start and stop the SIB and a viewer to visualize the content of the embedded SIB and has three tabs: Sessions, Subscriptions and Semantic model (to check insertions and removals among connected KPs and the SIB, key tool to test KP).

5. Future and conclusions

The ADK has already achieved its main goal, to aid developers programming high quality smart applications; nevertheless SOFIA aims to push this concept to the limit allowing final users to build their own applications. To do so, we are now defining a smart application model (within ODD) to design easy applications visually linking the different parts semantically [2],[3]. This approach will gain high dynamicity and composability inside a smart space, as the dependencies are not established statically but semantically and could be solved at runtime based on semantics. This feature will be ready with the second version of the ADK.

10. References

[1] W3C, Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Eng., 2006, online:http://www.w3.org/2001/sw/BestPractices/SE/ODA/.

[2] K. M. de Oliveira, K. Villela, A. R. Rocha, and G. H. Travassos, “Use of ontologies in software development environments,” in Calero, C., Ruiz, F., Piattini, M. (eds) Ontologies for Software Engineering and Software, 2006.

[3] M. Vanden Bossche, P. Ross, I. MacLarty, B. Van Nuffelen, and N. Pelov, “Ontology driven software engineering for real life applications,” in Proc. 3rd Intl. Workshop on Semantic Web Enabled Software Engineering, 2007.