1. Introduction

Nowadays ANDROID is the most popular mobile operating system in the world, and reaches out to touch peaks of diffusion, in some countries, more than 90% of the total smartphone market. The development of ANDROID together with the performance improvement offered by modern PDAs, like smartphones and tablets, has allowed many users to keep in touch with new kinds of applications that were not accessible to them in the recent past. Among them, applications related to the Agent Oriented Programming (AOP) paradigm [1] are particularly influenced by the wide diffusion of personal devices, thanks to their intrinsic mobile nature. In this paper we present a framework for programming agents in the ANDROID world, making it transparent to the programmer, basing it on the Knowledge Artifact notion.

2. Approach

While the artifact concept is often described as the result of a disciplined human activity, following rules based on training and experience, Knowledge Artifacts [2] are artifact specializations devoted to represent expert knowledge in object manufacturing or service delivery fields. The final aim of our Knowledge Artifact is generating executable rule-based systems written in JESS, minimizing the knowledge engineering effort [1]. To this scope, correlated tools for the representation of functional and structural knowledge (i.e. ontologies), procedural knowledge (i.e. influence nets) and experiential knowledge (i.e. task structures) have been integrated into a unique conceptual and computational framework, providing the user with an opportune set of primitives for designing and implementing decision support systems without a deep knowledge on the specific language syntax. This framework has been modeled as a client–server architecture, where the server is a knowledge-based agent having two tasks, the creation of the rule-based system according to the user (i.e. the expert) indications and the execution of it according to the data sent it by the client, that is a sort of reactive agent sending inputs and receiving outputs from the knowledge-based agent: in this way, it was possible to overcome the impossibility, at the current state of JESS implementation to import JESS library under the ANDROID OS.

3. Case study: the STOP system

This scenario has been successfully tested in a case study inspired us by the STOP handbook supplied to the Italian Fire Corps and Civil Protection Department of the Presidency of Council of Ministers for the construction of safety building measures for some building structures that have been damaged by an earthquake. After L’Aquila’s earthquake in 2009, this document has been prepared in order to standardize the steps to follow in similar situations. The structure of this document is suitable for the creation of a rule-based system, being modular and with specific and well defined cases. Using the created application, two knowledge models have been created (the first for the safety of the walls through rakers, the other for the safety of the openings through special scaffoldings). Every operator involved in the emergency procedures to make safe buildings and infrastructures is provided with an ANDROID application on his/her smartphone; this application has been modeled as a reactive agent, namely the STOP–Agent, communicating with the KA–Agent via the client–server architecture introduced above: Figure 1 (on the left) shows the interface for its initialization. Each STOP–Agent sends to the KA–agent data about the conditions of the site it is analyzing: according to the STOP handbook, these data allow to make considerations about the real conditions of the building walls after the earthquake in order to understand which raker (see the central part of Figure 1) or scaffolding (see the right part of Figure 1) to adopt. Exploiting the ANDROID primitives, it has been possible to create a stable mechanism for the communication with the server. The KA–Agent creates an instance of the KA model for each active STOP–Agents in communication with it. Then, it executes the model according to the rule–based system previously generated and sends answers to the STOP–Agent that will be able to take the proper action.

4. Conclusions

The proposed case study demonstrates the practical benefits coming from our framework: future works are devoted to transform it into a complete environment for developing and using decision support systems without the need of a knowledge engineer. This possibility could help people in proposing innovative applications which could be distributed over Internet and installed as apps on PDAs. Our targets are people traditionally out of the labour market, like young people and women, characterized by high levels of creativity and imagination. Android developers may find this tool suitable to develop “smart” systems, providing quickly solutions to a wide range of problems. To this aim, we are developing a new app for supporting a female hairdresser in her day-by-day activity, where she is acting as the expert producing the knowledge model.

References