Practical Applications of the Web-Based Agent Platform 'Eve'

Ludo Stellingwerff, Jos de Jong, and Giovanni E. Pazienza

Almende BV,
Westerstraat 50, Rotterdam 3016 DJ,
The Netherlands
{ludo,jos,giovanni}@almende.org

Abstract. The existing approaches to build multi-agent systems fail at addressing the challenges posed by the current technology, where ubiquitous interconnected electronic devices are no more passive machines operated by humans but rather active computational components cooperating with humans. In order to tackle this problem, we have created a novel open-source web-based agent platform called 'Eve' that features some specific characteristics (e.g., platform and language independence, openness) that make it particularly suitable to be deployed in real-life applications. In this paper, we discuss the main features of Eve and present several use cases in which it has been successfully applied.

Keywords: agent platforms, multiagent systems, interoperability.

1 Introduction

Nowadays, both humans and software applications can be considered as entities with some degree of autonomy that interact with each other and with the environment without need of centralized coordination. In this framework, devices are modelled as agents that mimic some of the characteristics of human beings: they are autonomous (i.e., capable of taking decisions), intelligent (i.e., capable of adapt their behaviour on the basis of available data), and social (i.e., capable of communicating with humans as well as other agents). Therefore, the fundamental tool to handle (and profit from) such a complex yet promising scenario is to build an effective multi-agent system (MAS) [11] tailored for this novel technological context. A MAS is usually developed on an agent platform, which is usually chosen among the several dozens already available on the market (a good yet not recent overview can be found in [13]). Nevertheless, the great majority of them – usually Java based – consist of a closed and controlled environment (operating system or simulation environment) where agents can live and interact with each other. In general, these platforms suffer from connected to the scalability and to the robustness, mainly because of the presence of central directory services (which require memory to be stored) and to the amount of manual setup to add new agents, especially when the platform runs in a heterogeneous environment.

Y. Demazeau et al. (Eds.): PAAMS 2014, LNAI 8473, pp. 268–278, 2014.

In order to overcome these problems, we propose 'Eve' [8] that is an agent platform specifically thought to be deployed on a diverse distributed environment. Eve is inspired by the principles of human-agent collectives described CHAP [17] and in ORCHID [15], in which agents (both human and software) collaborate in a seamless and effective way. Eve has been the key component of several successful applications, especially in the fields of emergency management, smart grids, energy consumption optimisation, and coordination of complex tasks. In this paper, we also discuss several of these practical use cases, emphasising that they are just a few of those in which such innovative agent platform may find application.

The paper is structured as follows: in Sec. 2, we describe the architecture and the main features of Eve; in Sec. 3, we illustrate a few practical examples in which Eve has already found application; in Sec. 4, we draw the conclusions of our work.

2 Summary of the Main Features of Eve

2.1 Conceptual Architecture

From the architectural overview of Eve shown in Fig. 2, it is possible to catch a glimpse of its main features. It should be evident that Eve can be deployed on a number of different devices (smartphones, servers, local PCs, etc.) hosting different environments (JavaScript, Android, C++, etc.), all connected to the internet via the JSON-RPC protocol.

Thanks to this approach, Eve has some distinctive characteristics. First of all, Eve is platform independent since agents can live on any device: smartphones, robots, servers or, more generically, in the cloud. Second, Eve is language independent[1] because it dictates only the communication protocol (JSON-RPC) which works over existing transport layers (HTTP, XMPP) and a simple API, as described in and Table I. Third, Eve is an open agent platform: each agent has its own public[2] URLs and hence existing Eve-systems can be easily connected to the others. Furthermore, non-Eve systems can be connected to the Eve platform by making its API available via an Eve agent acting as a wrapper.

The architecture of Eve leads to further advantages among which is worth to emphasize the following ones:

Scalability: Eve is fully web-based and hence, from a practical point of view, there is no upper bound to the number of new agents that can be added to the system without degrading its performances.

Robustness: the state persistency in Eve is offloaded from the agents to the environment (i.e., the states of the agents are distributed), which makes Eve

[1] Currently, there are two mature implementations of Eve (one in Java and the other in JavaScript) whereas a third one (in Python) is in an embryonic stage.

[2] Authorization mechanisms have already been implemented, in case the application needs them.