

Persistent Architecture for Context Aware Lightweight Multi-agent System

Aqsa Bajwa¹, Sana Farooq¹, Obaid Malik¹, Sana Khalique¹,
Hiroyuki Suguri², Hafiz Farooq Ahmad², and Arshad Ali¹

¹ NUST Institute of Information Technology,
166-A, Street 9, Chaklala Scheme 3, Rawalpindi, Pakistan
Tel.: +92-51-9280658; Fax: +92-51-9280782

{aqsa.bajwa,sana.farooq,obaid.malik,sana.khalique,
arshad.ali}@niit.edu.pk

² Communication Technologies,
2-15-28 OmachiAoba-ku, Sendai 980-0804 Japan
Tel.: +81-22-222-2591; Fax: +81-22-222-2545

{suguri, farooq}@comtec.co.jp

Abstract. Application development on handheld devices using software agent technology is becoming more and more popular around the world. Escalation in the use of lightweight devices and PDA's leads us to create a concrete base for future nomadic applications, positioned in a changing environment. However, constrained characteristics of handheld devices serve as the main hindrance towards achieving this goal. This paper presents the architecture of context aware FIPA compliant multi agent system for the lightweight devices called SAGE-Lite, which are capable of providing fault tolerance through the mechanism of object persistence. Agents existing on lightweight devices can communicate and provide services via Bluetooth and the communication with the server is done via WAP. Since agents communicate via sending and receiving ACL messages, this architecture will minimize communication latency with in the platform. This framework allows implementing agent-based applications like business applications or e-commerce applications on these resource-constrained devices.

1 Introduction

Designing and implementing complex software systems have always been much complicated and time consuming. Different research communities have been striving to achieve better methods and techniques for the development of such systems and this struggle led to the basis of software agent technology, which a few years back became an active research area not only for academia but commercially as well. The research on wireless communication technology has also become popular lately as nomadic computing; wireless data communications and mobile devices enable accessing fixed network services from almost anywhere and any time [1].

Software agent technology is a capable approach for the analysis, specification and implementation of complex software systems. Multi-agent systems are systems composed of multiple agents, which interact with one another, typically by exchanging messages through some computer network infrastructure. MAS provide proper execution environment to agents so that they can assure the provision of services to other agents by cooperating, coordinating, and negotiating.

It is about time that these two interesting and important technologies, nomadic computing and software agent technology, start to converge [1]. About 1/3 of the world's total population is mobile Internet users as handheld device are becoming more than a personal information system. So the communicative agent technology in our opinion will reach new heights in this direction.

Foundation for Intelligent Physical Agents or simply FIPA is one of the standard governing bodies [3], which provide an abstract architecture for lightweight multi-agent system developers to follow. Agent Management System (AMS), Agent Communication Language (ACL), and Message Transport Service (MTS) are one of its mandatory components. Figure 1 shows the list of mandatory and optional of lightweight MAS.

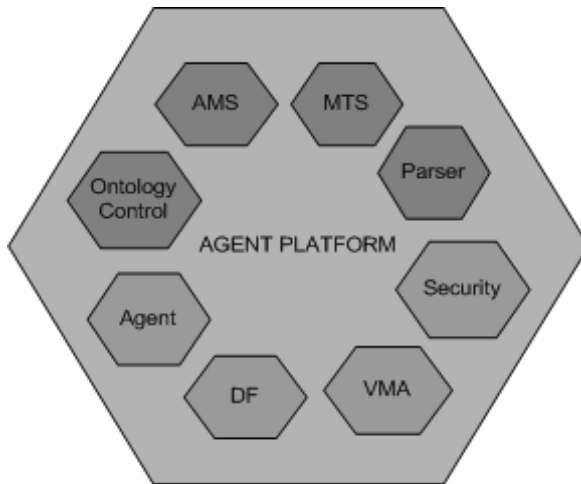


Fig. 1. The mandatory and optional components of a lightweight agent platform

Wireless devices have gained a lot of significance in the past few years. Diversified ranges of mobile applications are becoming popular day by day. The applications run using the constrained resources provided by today's vast variety of the cellular devices.

The resources in a handheld device are constrained in terms processing power, memory, graphical user interface support, screen size and permanent storage. On the network side, wireless networks are constrained by less bandwidth, more latency, less connection stability and less predictable availability.