Mobile agent based framework for mobile ubiquitous application development

Seungkeun Lee • Kuinam Kim

Published online: 25 March 2011
© Springer Science+Business Media, LLC 2011

Abstract Ubiquitous computing which enabled by the availability of mobile, heterogeneous devices that supply context information, is currently not matured by the lack of programming support for the design and development of context-aware applications. Especially, ubiquitous computing environment is not static which can be compensable dynamically according to need of environment. Mobile agent is a very efficient framework applications can cooperate in heterogeneous environment. Therefore, we have developed a mobile agent based framework that significantly eases the development of mobile, context-aware applications. The framework allows developers to fuse data from disparate sensors, represent application context, and reason efficiently about context, without the need to write complex code. An event based communication paradigm designed specifically for ad-hoc wireless environments is incorporated, which supports loose coupling between sensors, actuators and application components.

Keywords Mobile agent • Mobile ubiquitous computing

1 Introduction

This Context awareness and mobility are core concepts in the vision of ubiquitous computing where networks of small computing devices are dispersed in the physical environment, operating autonomously and independently of centralized control. Context-aware applications are a large and important subset of the overall set of ubiquitous computing applications, and have already demonstrated the advantages gained from the ability to perceive the surrounding environment. Such applications however remain difficult to develop and deploy, with no widely accepted programming model available. Programmers are often required actuator devices at a low level in order to develop relatively simple applications [1–3].

The mobility of devices in the ubiquitous computing environment also raises challenges in the areas of communication and interaction due to factors such as dynamically changing network addresses and system configurations, susceptibility to disconnection and low bandwidth [4].

The main components of a context-aware application are a set of sensors for capture of context data, a set of rules governing behavior according to context and a set of actuators for generating responses. We have developed the sentient object model for the development of context-aware applications in an ad-hoc mobile environment, which defines software abstractions for sensors and actuators, and provides a framework for the specification of production rule driven behavior. Sentient objects have a number of characteristics that are important in ubiquitous computing environments:

- Sentience—the ability to perceive the state of the environment via sensors
- Autonomy—the ability to operate independently of human control in a decentralised manner
- Proactiveness—the ability to act in anticipation of future goals or problems.

The framework fulfills the two major goals identified by Dey and Sohn [5] that are necessary for the successful development of ubiquitous, context-aware applications, namely:
Applications are easier to design, prototype and test, supporting a faster iterative development process. Designers and end-users are empowered to build their own applications. A number of other middleware proposals address the challenges of effectively developing context-aware applications.

Seminal work by Dey and Sohn [5] provided a toolkit which enabled the integration of context data into applications, but did not provide mechanisms for performing sensor fusion, reasoning about context, or dealing with mobility. Context acquisition and use was often tightly integrated into a single application [2], and could not easily be incorporated into other applications. Other work provided mechanisms for reasoning about context [1, 6], but still did not provide a well-defined programming model and did not address the challenges of mobility. Recent and ongoing work [7–9] provides programmer support for the development of context-aware applications, but does not provide the ability to systematically specify and manage event filtering, sensor fusion and rule-based inference in a mobile ad-hoc environment, as our framework does.

2 Common requirements of context aware frameworks

From the functionalities of the frameworks studied above, we came up with the common set of requirements that any context aware framework satisfies:

- Sensor technology to capture the contextual information: Acquire raw contextual information
- Support for event based programming model so as to have the ability to trigger events when certain context change is observed.
- A way to communicate the sensed contextual data to other elements in the environment and a way to interpret the collected data: provide interpreted context to application.

Support to build context aware applications is provided by various infrastructures indicated above among which the two promising supporting infrastructures for our thesis are: The Context Toolkit and the Java Context Aware Framework. JCAF would have been a better choice for our thesis, since it supports an Application Programming Interface (API), which can be used directly to handle contextual changes and since it is based on deployment in an organization. However, it is relatively new and is under active development. Being a relatively new proposal, JCAF has no proper documentation as of now. Context toolkit is relatively well established and the first version is now publicly available. Hence, modified context toolkit is a better choice for use in the framework proposed in this thesis.

3 The sentient agent programming model

We provide a programming model, based on the sentient object model and incorporating the STEAM event service,