Encapsulating the environment: a flexible framework to develop ubiquitous applications for intelligent environments

Matheus Erthal
Institute of Computing
Federal Fluminense University
Rio de Janeiro, Brazil
Email: merthal@ic.uff.br

David Barreto
Institute of Computing
Federal Fluminense University
Email: dbarreto@ic.uff.br

Douglas Mareli
Institute of Computing
Federal Fluminense University
Email: dmareli@ic.uff.br

Orlando Loques
Institute of Computing
Federal Fluminense University
Email: loques@ic.uff.br

***Abstract*—Due to recent advances in mobile computing and wireless communication technologies, we can see the emergence of a favorable scenario for building ubiquitous/pervasive applications. This work proposes a new framework that aims at building those applications, providing a set of concepts and implementation tools. We propose abstractions that allow developers to handle the resources spread in the environment in a simple and homogeneous way, and to interpret context information. In order to demonstrate the feasibility of the proposal, the concepts were implemented in a platform named *SmartAndroid*. A ubiquitous applications prototyping interface was implemented over this platform to allow testing of different environment configurations before purchasing all devices; and also a context rules composition interface, where end users can define their preferences in the environment.**

I. Introduction

The field of Ubiquitous Computing (UbiComp), nowadays widely discussed, was first proposed by Mark Weiser in the 1990's [1]. Also referred by pervasive computing, the field aims at providing a different paradigm of human-machine interaction. As traditional applications provides services to users through their explicit interaction (using devices as mouse, keyword, monitor, for example), in ubiquitous applications the interaction happens without the need of explicit interaction, i.e. the application tries to discover the users needs through the acquisition of context (using sensors) and the knowledge of their preferences, and providing services in the environment (using actuators). These environments enhanced with sensors and actuators to provide automated services to persons are also called Intelligent Environments (IE).

The construction and manipulation of ubiquitous applications represent major challenges for developers, especially in terms of technical knowledge required and the availability of real devices during application development. Some of these challenges can be well highlighted: (i) there are difficulties in establishing a common protocol for communication between the components of the distributed system, because of the *heterogeneity of devices* involved, (ii) the interactivity of ubiquitous applications is hampered depending on the amount and *variety of context information and services* available in the environment, (iii) *developing and testing applications* require high availability of resources, such as sensors (e.g. presence, lighting, temperature), actuators (e.g. keys, alarms, smart-tvs), including new embedded devices, or physical spaces, such as a house for applications of type smart home.

In this work we propose a framework for developing ubiquitous applications in IE. The goal is to provide support for the programming, testing and execution of applications, thus allowing to deal consistently with systems great complexity. The framework stands out for addressing the challenges already identified in UbiComp [2]. The *heterogeneity of devices* is handled through the definition of a Distributed Component Model, that provides abstractions to encapsulate these devices, also called resources, enabling developers to interact with them seamlessly. Regarding the *variety of context information and services* issue, we propose a solution for context interpretation that allows developers to create and manage context rules at runtime, and users to set their preferences in the IE. Finally, regarding the issue on the *resource availability*, the framework includes an application interface for the prototyping and management of pervasive applications (IPGAP), focused on visualization and testing ubiquitous applications, mixing real and virtual components.

The concepts of the framework were materialized on a platform called *SmartAndroid*[1], whose development has enabled appraisal in order to prove that the conceptual framework facilitates the process of building ubiquitous applications. We implemented some use cases to validate different aspects of the framework proposed and to testify its capabilities, supported by the elaboration of competency questions, which is a mechanism mostly used to evaluate ontologies, but can also be used as an assessment to this work.

The remainder of this paper is organized as follows. First we present in Section III an overview of the main concepts used as a basis for the framework's development. In Section IV, we present the framework's overall architecture and main features. We then, in Section V, present a proof of concept demonstrating the feasibility of building ubiquitous applications using the framework and we show examples of applications that explores the key features of the IE. Finally, in Section II we compare our approach with related work and in Section VI we conclude this paper with the main remarks.

[1] www.tempo.uff.br/smartandroid

II. Related Work

The benefits to users and developers that UbiComp foresees reach beyond the horizon of many imaginative researchers. The possibilities that arise from it surpass the bounds established by keyboards and mice, our standard interfaces. Nonetheless, building such adaptable applications requires much effort from developers that see themselves immersed in a universe of device's specifications and communication technologies, apart from the problem itself that the application must solve.

Therefore, many researchers have focused on diminishing those obstacles by providing abstractions, middlewares, services, tools, and other suportive techniques, not only for development but also for prototyping.

As one of the most famous, the Gaia middleware is designed to facilitate the construction of applications for IE. It consists of a set of core services and a framework for building distributed context aware applications. The main goals of Gaia's approach are the acquisition of context by applications, the monitoring of entities location, the maintenance of hardwares and softwares descriptions, and

III. Overview

It is easy to notice that the context information is a first concern topic in ubiquitous systems. But what is "context"? Many authors have proposed definitions to this concept, though most are little accurate [3]. Synthetizing previous definitions, Dey and Abowd have proposed that context is "any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves" [4]. In a IE "Places" are the rooms, the floors of a building, or spaces in general that enable the localization of other entities, "People" are individuals that populate the environment and interact with it, and "things" are virtual representations of physical objects or software components.

Context aware (or context sensible) applications is a designation to those applications that not only are capable of knowing the context of the environment, but also react to it either by means of changing the environment or in a software level.

[distribs sys]

IV. Framework

[framework description]

A. Resource Agents

[communication mechanisms]

B. Architeture

[distributed components model]

C. Management Services

[resource register service]

[resource discovery service]

[resource location service]

[location concerns]

[security concerns]

D. Context Interpretation

[directly programmed]

[as a separated entity]

E. Framework Implementation

[android and other techs]

[impl of AR and services]

[impl of context rules]

[context rules conflicts is an ongoing study that aims at discover a conflict in runtime,]

F. Management and Prototyping Interface

[description]

[main figure]

[rules composer]

V. Case Study

VI. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 3, pp. 94–104, 1991. [Online]. Available: http://wiki.daimi.au.dk/pca/_files/weiser-orig.pdf

[2] R. de Araujo, "Computação ubíqua: Princípios, tecnologias e desafios," in *XXI Simpósio Brasileiro de Redes de Computadores*, vol. 8, 2003, pp. 11–13.

[3] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.

[4] A. Dey, G. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human-Computer Interaction*, vol. 16, no. 2, pp. 97–166, Dec. 2001. [Online]. Available: http://www.informaworld.com/openurl?genre=article&doi=10.1207/S15327051HCI16