A preemptive ubiquitous system

Bruno R. Nunes, Guilherme A. Borges,
Valderi R. Q. Leithardt, Cláudio Fernando Resin Geyer
Informatics Institute
Federal University of Rio Grande do Sul
Porto Alegre, RS, Brazil
{brnunes, gaborges,
vrqleithardt, geyer}@inf.ufrgs.br

Abstract

This paper presents the project of an ubiquitous system that preemptively manages features of the environment, according to predefined preferences for the users currently present in it. Using multiple location technologies, the system detects when users enter or leave an environment, and adapts attributes of the environment to fit their preferences. The prototype focus on an Android application that manages settings of the device, changing the state of some functionalities based on the location of the user.

1. Introduction

Computers are an incredibly recent technology, that evolved from old calculator machines and became essential on people's lives. They started big and expensive, with the purpose of helping on heavy calculations, and are now cheap, small and powerful enough to be added everywhere we need. Thanks to that progress, we are living today the process of proliferation of the computers and their fade to the background, as predicted by Mark Weiser back in 1991 [1].

Every day more and more computers are added to objects we use on our day by day. Often we did not imagine these items could get better at their function, but they are being improved with the help of computers that connect wirelessly.

Televisions now have complex operating systems that support online video streaming, video conferences and internet surfing; refrigerators include LCD screens and run custom applications; doors locks are controlled by computers; cars include complete computers that are used to control functions of the car, navigate, or even to provide entertainment. Every object is getting equipped with sensors and connected to the internet. It's the growing "Internet of Things".

These examples are all of products that are already established. There are still lots of concept products being launched regularly from both academical and commercial research that embed computers into regular objects to give them intelligence and connectivity. Although at this time they are only ideas, it is safe to predict that a lot of them will eventually be absorbed by society.

On the same trend, smartphones are becoming essential in people's lives. Even though very recent in existence, they are already deeply pervaded over the world. They are powerful computers that are always on and accompanying their owners. They are also very personal, as users customize them and fill them with personal information and data. Together with their reduced size and constant connectivity, this makes such a device perfect for use as an identification of its owner, allowing the development of context-aware computing around it.

With this work, we want to create a system that integrates these diverse computers and make them work preemptively, according to the environment and the people currently in it. Our final goal is to create the basis for an expansible system that manages ubiquitous computing environments. The system is intended to integrate the multiple computers of the environment, and administer a network where each computer is responsible for the control of some feature of the environment.

Using localization sensors, the system will be aware of the presence of users in the environment, and will adapt it to the preferences defined for the people that are currently present. To accomplish that, a model of the system will be designed, and a prototype will be developed. The prototype consists of an implementation of the complete model, with focus on an Android application.

We will take advantage of one of the most powerful features of modern smartphones: the presence of multiple embedded sensors. The Android application will use multiple location technologies to detect the presence of the user in the covered environments. The application will also turn the device into part of the environment by automatically managing functions of the smartphone according to preferences defined for the environment.

2. Proposed Model

2.1. Terminology

To facilitate the understanding of the ideas developed in this work, a terminology was created and will be used along it:

- **User**: Any person registered on the *system*, that will interact with the *environments*.
- **Environment**: A specific location, of any magnitude (a campus of a university, a shopping mall, a building or a room) that is part of the *system* and will interact with the *users* present in it, based on predefined actions relating the environment with these *users*.
- **Feature**: An attribute of the *environment*, controlled by an *action client*, that therefore can be modified responding to messages from the *server*.
- **Functionality**: An attribute of a device that runs an *action client application*. It differs from a *feature* in the sense of being an attribute of the *client*, not of the *environment*.
- **Action**: A setting operation over one *feature* or *functionality*. Actions are created by the *server* based on the current location of the *users*, and sent to an *action client*, that will execute it.
- **Main Database / Server Database**: Database modeled and implemented to support the *system*. Comprehend all the variables taken into account on the logic that the *server* uses to define the *actions* to be taken on any given situation.
- **Server**: The application running on a server machine that manages how the system works. It is always available for contact from the *localization client applications* that send updates on the presence of registered *users* in tracked *environments*. It is also responsible for generating the *action* messages and sending them to the appropriate *action clients*, that will execute them.
- **Client / Client Application**: General term referring to an application that interacts and is subordinated to the *server*.
- **Localization Client**: Client applications that are responsible for keeping the *server* informed about what *users* are present in the tracked *environments*.
- **Action Client**: Client applications that manage one or more *features* and *functionalities*, and therefore are responsible for the execution of *actions* over those.
- **System**: The system as a whole, composed of *database*, *server* and *clients* working together to create the ubiquitous *environment*.
- **Device**: Any machine that runs an *action client application*, or is controlled by a computer that runs it. Devices have control over *features* and *functionalities*, being the physical executors of the *actions* generated by the server.

2.2. Model

The system implements a client-server architecture that communicates over the internet. It is composed of multiple client devices and a server backed by a database.

The client applications can be grouped into two types, according to their contribution to the system. The first group embraces the *Localization Clients*, that use one or more localization technologies to determine the presence of users in covered environments. The *Localization Clients* can be part of an environment, and detect when users are in its bounds, or be associated with a user and detect in which environment the user is at a given moment.

The second group of client applications is composed of *Action Clients*. *Action Clients* are those that manage one or more functionalities related to the environment. They are accountable for changing the state of these functionalities upon receiving action messages from the server.

The server is the core of the system. It receives presence information from the *Localization Clients*, defines what actions should be taken, and send messages to the *Action Clients* so that they can execute that actions. The reasoning to define the actions is based on a database. The database has information about every user and environment of the system, and rules associating them. These rules are analyzed at every change of context and transformed in actions that the *Action Clients* can execute, changing aspects of the environment.

Figure 1 illustrates how these components interact. The Android client application is included in the *Action Client* and *Localization Client* groups. It locates the device's owner using NFC and the Android Location Provider, and manages device's functionalities based on where its owner is at the moment.

3. Related Work

Subsection 3.1 presents studies about location that could be integrated to this work as localization client applications. Subsection 3.2 presents Android applications on the market

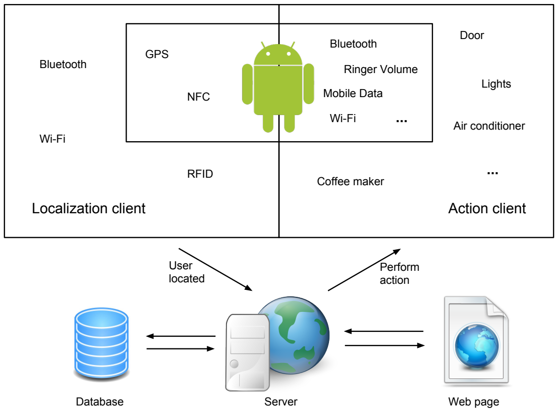

Figure 1. System model overview

that function in a similar way to our Android client prototype.

3.1. Localization

Diverse studies propose ways of improving the capabilities of the existent technology as means to effectively locate objects in the space. If those ideas are implemented following the communication interface we defined for our system, they could be integrated to it and improve its overall performance.

The use of a robot equipped with a RFID reader is the proposal of [2] to improve the precision of RFID systems. While in the traditional use of RFID the reader is supplied with constant power, in this study the idea is to vary the reader's power to detect the distance of the tag from the antenna. First, the maximum power is set to sense tags in a long-range. Then the power is reduced in steps, which also reduces the sense range of the reader to known bounds. That allows the system to estimate the position of the tags in the environment.

The development of an indoor localization system for Android mobile devices is presented in [3]. Relying on the sensors of the devices, the solution merges three localization techniques to improve accuracy: received signal strength fingerprint, dead reckoning and Sequential Monte Carlo filtering. It displays the user's location on a floor map, and other layers can be added to show points of interest.

Another solution that provides Android devices with the capability of indoor localization is [4]. The objective is to bring location based services to the indoor level, in environments such as shopping malls, museums, exhibition centres and airports. Since GPS is not a good solution inside buildings, the Airplace system uses the received signal strength data from the surrounding WiFi infrastructure to locate the user. It takes advantage of the WiFi system that is usually already deployed in the environment and creates a "radiomap" of the place, that is then used by the Android application to locate the user indoor through state-of-the-art algorithms.

3.2. Android Automation Applications

The idea of automating tasks on Android devices is not new. Browsing the Google Play store [5], we can find a number of applications that promise to automate actions on the occurrence of some event. The most popular are AutomateIt [6], Tasker [7] and NFC Task Launcher [8]. The basic idea behind these applications is to allow the user to setup rules that define actions to be executed on the occurrence of some event. By doing so, users can program their devices to automatically act on situations where they would execute a task manually.

These applications generally offer a numerous set of events that can be used as triggers. Among them are: a date or time of the day; the state of some hardware of software functionality, such as the Wi-Fi or Bluetooth, network signal strength, battery level and battery state (charging, discharging); the location of the device; a received call or SMS; etc.

The actions can also be chosen from an extensive set, that includes but is not limited to: display some kind of alert; change the state of some functionality, such as mobile data, Wi-Fi, Bluetooth, autosync, screen brightness; launch an application; get location; set volume; etc.

Like the researched automation applications, the Android application developed on this work automates the execution of tasks on the device it is installed. The difference begins with the fact that these applications respond to local triggers, while ours is a client that responds to messages from the server indicating what it should do.

Instead of using the state of the device and events that happen on it, our application executes actions based on the relationship between the owner of the device and the environment in which it is currently located. The actions are managed by the server, which knows about the location of the user and rely on rules defined by the administrators.

Table 1 compares characteristics of the researched apps and this work.

4. Prototype

The Android client application is being developed to validate the proposed model. It integrates with the server implementing both *localization client* and *action client*.

Due to the different levels of accuracy needed on the location of users, two *localization clients* are being developed and will work concurrently on the application. The first uses the Android Location Services, which relies on the sensors of the device (GPS, wireless network information and cell

	AutomateIt, Tasker and similar	NFC Task Launcher	This work
Where the actions are defined	Locally	NFC Tag	Server
What triggers an action	Events on the device	Reading of NFC Tag	Presence detected in environment
Use on multiple devices	Actions must be created for each device individually	Any device can read the tag and execute the actions	New devices can be added to the system and actions will be assigned to them according to rules defined for their owners

Table 1. Comparison between existing apps and this work

tower ids). The positions gathered by this means can determine the location of the user up to a building level. However, they are not precise enough to determine in which room the user is. To locate the user more precisely in the environment, the second *localization client* implemented uses NFC. With NFC it is possible get a more precise position, given that we know exactly where the NFC tag is placed, and that it is only possible to read the tag within a few centimeters from it.

The Android client application also implements an *action client*, that changes the state of some functionalities of the device upon receiving commands from the server.

5. Conclusion and Future Work

With the development of the prototype we expect to achieve a completely functional system, allowing us to measure the quality of the proposed system.

Future works include the development of action clients that control features of the environment, such as a the air conditioning or lighting system. Those clients should comply with the communication protocols we defined in this work and would be easily integrated with the system, enabling a better customization of the environment. The addition of more localization clients would also improve the performance of the system, as we would be able to get user's postitions in more quantity and quality.

References

[1] M. Weiser, “The computer for the 21st century,” *Scientific American*, vol. 265, pp. 66–75, Sept. 1991.

[2] T. Hori, T. Wda, Y. Ota, N. Uchitomi, K. Mutsuura, and H. Okada, “A multi-sensing-range method for position estimation of passive rfid tags,” in *Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*, WIMOB ’08, (Washington, DC, USA), pp. 208–213, IEEE Computer Society, 2008.

[3] J. Sevcík, “Indoor user localization using mobile devices,” *ERCIM News*, vol. 2013, no. 93, 2013.

[4] C. Laoudias, G. Larkou, D. Zeinalipour-Yazti, and C. G. Panayiotou, “Airplace: Indoor geolocation on smartphones through wifi fingerprinting,” *ERCIM News*, vol. 2013, no. 93, 2013.

[5] Google, “Google play,” 2013. Visited May 2013.

[6] AutomateIt, “Automateit - make your smartphone smarter!,” 2013. Visited May 2013.

[7] Tasker, “Tasker - total automation for android,” 2013. Visited May 2013.

[8] Tagstand, “Nfc task launcher,” 2013. Visited May 2013.