Workshop Proceedings of the 6th International Conference on Intelligent Environments. Ambient Intelligence and Smart Environments, Volumen 8. IOS Press, 2010. 42-51

DOI: http://dx.doi.org/10.3233/978-1-60750-639-3-42

Using 2D Codes for Creating Ubiquitous User Interfaces for Ambient Intelligence Environments

Javier Gómez, Germán Montoro, Pablo A. Haya, Xavier Alamán
Universidad Autónoma de Madrid
{jg.escribano, german.montoro, pablo.haya, xavier.alaman}@uam.es

Abstract. Smart phones are one of the most popular devices nowadays. The enrichment of their technical capabilities allows them to carry out new operations beyond the traditional in telephony. This work presents a system that automatically generates user interfaces for Ambient Intelligence environments. This way, smart phones act as “ubiquitous remote controllers” for the elements of the environment. The paper proposes some ideas about the usability and adequacy of these interfaces.

Keywords. Ambient Intelligence, Ubiquitous User Interface, Mobile Device, 2D Code

Introduction

Ambient Intelligence provides new possibilities of interaction [1] offering new challenges to the designers of the interfaces [2]. These new interfaces should be context aware and should adapt to the environment, the users and their needs.

For this purpose, we have created the Ambient Intelligence Laboratory (AmILab) at the Escuela Politécnica Superior of the Universidad Autónoma de Madrid. This Laboratory is furnished as a living room with TVs, couches, small electrical appliances, and also sensors and actuators. [3]

The interaction with the elements of the AmILab environment is possible thanks to an oral interface [4] and a graphical interface [5], by means of direct control. In the case of indirect control, the system counts on a mechanism based on rules and agents [6].

From the direct control point of view, the interaction with the environment was limited to the PC. This is enough in many cases, but in some situations it could also be desirable to interact with the environment using another kind of devices. (i.e. mobile devices or smart phones).

Mobile devices (smart phones, PDAs, etc.) are becoming very popular and their technological capabilities grow every day. Current devices can perform numerous tasks, thanks to its memory and processor improvements, new hardware components (High definition cameras, GPS receivers, digital compass, etc.) and, also, some of these new capabilities are provided thanks to the new operating systems (iPhone OS, Android, Windows Mobile, etc.) The connectivity has also been improved since they are able to connect to 3G, HSDPA or WiFi networks.

In this work we propose a system that automatically generates ubiquitous user interfaces to control the environment. Our aim is to study the use of smart phones as "ubiquitous remote controllers" for the elements of an environment. In the same way we point at the TV with the remote to perform an action, i.e. changing the channel, we suggest recognizing the element by the use of 2D Codes. Thus, we can identify the element we are pointing with the phone camera and, then, generate an adapted and personal interface to control it.

Ubiquitous interfaces offer some advantages, such as adaptation and mobility. The adaptation is not only related to the device and its constraints, but also to the environment, the context, the users and their preferences and special needs. Regarding the ubiquity and mobility, as these interfaces are executed on a mobile device, it is guaranteed. Since these interfaces are generated automatically, any user could be able to control any new device from the first time although she did not have any technical knowledge or special skill.

This article is structured as follows: In Section 1, we give an overview of the related work: user interfaces for mobile devices. In Section 2 we will talk about the proposed system, the architecture, design steps and implementation. Finally, in Section 3 we show the results of a usability evaluation study; in Section 4 we conclude and, in Section 5, we present some lines of future work.

1. Related Work

In the literature, we can find many different projects oriented to generate user interfaces from the information stored in 2D codes. Some of them are considered marker based augmented reality systems.

Rekimoto, J. *et al.* [7] presented one of the first works that used 2D codes. In this article the authors describe a set of examples where 2D codes were used as augmented reality tracking marks. These were: links to the digital information, an indoor navigation system, 3D annotations over the real world, direct manipulation of objects of physical environments, TV shows' ID transmission and recognition of objects in ubiquitous computing environments.

Wagner and Schmalstieg proposed in [8] one of the first augmented reality systems for mobile devices. This project consisted of a 3D navigation system for PDAs for guiding people in a building. Because of the limitations of PDAs in 3D calculus, they implemented an assisted version, based on the client–server architecture. The server made the 3D calculus relative to the tracking, while the PDA performed the tracking itself (with ARToolKit [9]) and the presentation of the data.

Another work form Wagner *et al.* [10] presented a system architecture for interactive, infrastructure-independent and multiuser augmented reality for mobile devices. This architecture was based on the framework Studierstube [11]. It also used some other tools, such as ACE [12] for network communication and KLIMT[1] for graph representation. As a demonstration of the architecture, they presented the application "The invisible train". In this multiuser game, players had to guide virtual trains over a scaled model real track. Players used PDAs equipped with cameras to "see these invisible trains". Since the scaled model was tagged properly with visual tags, the PDAs that players used were capable to determine the position of the observer and the

[1] Open Source 3D Graphics Library for Mobile Devices. http://studierstube.icg.tu-graz.ac.at/klimt.

track, so the system was able to “draw” virtual trains over the image captured by the camera.

Looking for systems or projects in which 2D codes were used for data storing instead of being used as visual marks, we found the work proposed by Siltanen and Hyväkkä in [13]. This system tried to avoid the user to introduce large strings by storing all that information in a 2D Code. Once the 2D Code was read, an operation was suggested to the user: opening the browser if the information contained an URL or making a phone call if the information was a phone number. They also made a study to try to establish a relation between the size of the code, the amount of information that it could store and the maximum reading distance. They concluded that two ways of reading were necessary: a fast but low resolution one and another one slower but high resolution.

Al-Kahalifa presented in [14] a system that describes objects that had been previously tagged with QR Codes to assist visually impaired people. The user had to point at the object with her cell phone (that has to be equipped with a camera) and, automatically, the device recognized the QR Code and asked an external web server for an oral description of the object. In this way some extra information was given to the user. The system was tested in two different scenarios: the first one was a museum, in which some of its pieces were tagged. The other one was a test that tried to describe products (in a shop, for example).

2. Proposed System

The proposed system is based on the “ubiquitous controller metaphor” as its interaction mechanism. Its objective is to generate multiple adapted interfaces to control a wide variety of devices (even unknown by the user) at any time.

To identify the devices that a user can interact with, the elements that compose the environment are tagged with 2D Codes. If a user points at one of these codes with the camera of her mobile device, it automatically scans the code and extracts the information stored in it. Then the mobile device automatically generates an adapted interface that allows interacting and controlling that element of the environment.

A supporting architecture is required to generate interfaces automatically. This work has been developed using the blackboard architecture described in [15]. This blackboard acts as a middle layer between the environment and the applications, so the access to the physical devices is easier for the programmers. The architecture has also the capability of storing generic interface descriptions, which are employed for the generation of the mobile user interfaces.

This blackboard stores entities that may be composed of properties, capabilities and relations with other entities. It is based on an ontological model, so entities are realizations of the classes described in the ontology. Therefore, an entity will have also a type (that matches up with a class described in the ontology), a name (unique) and a domain (there is a domain’s hierarchy to facilitate the scalability of the system).

On the other hand, we think control interfaces should be based on the actions that an element can perform. Since these actions are represented in the ontology as capabilities, the control interface is based on these capabilities. Even more, as capabilities might be composed of properties, each of these properties is responsible for the interface that will be shown to the user.

Finally, in view of the different kinds of 2D Codes [16] and the nature of the information they will have to store, it seemed useful to choose an index oriented code but, given the absence of standards and the architecture that the system is based on, data oriented codes are preferable. More specifically, QR Codes are the selected ones since they are standard and, at the moment, they are free for use. There are also free generators[2] and libraries for the capture and decode processes[3].

QR Codes can store the amount of information necessary to retrieve the entity from the blackboard. That is the “Full Name” of the entity, which is composed of three parts: the type, the name and the domain of the entity. From this “Full Name”, we can extract all the information from the blackboard and generate the interface automatically.

From a development platform point of view, we have chosen Android. It is an open mobile system that is increasing its market share everyday and the devices that run it have the necessary hardware to execute the application.

The implementation is divided into four parts or activities. Each of these four activities corresponds to an action that the user must participate on: system access, initial selection, QR Code scanning (capture and decode) and device controlling.

We can classify these activities into two groups: static and dynamic. Both access and initial selection activities and QR scanning are static, namely they have always the same structure, elements (buttons, etc.) and perform the same task. On the other hand, the activity that presents the interface that allows the user to control the device is dynamic. It changes in execution time: it is different for different entities and, even for the same entity, it can change. A diagram of the activities of the application is shown in Figure 1.

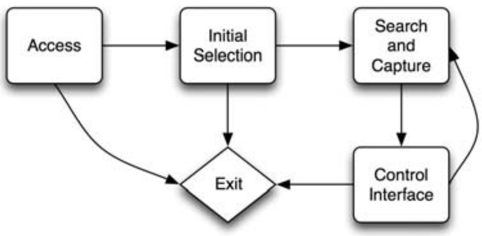

Figure 1. Activities diagram of the application

Once the QR Code is scanned and decoded, the activity “Control Interface” starts. From the “Full Name” of the entity, extracted from the QR Code, the system is able to ask the blackboard for the representation information associated with that element. This information is stored in the blackboard as entities’ properties and also as capabilities’ properties, which are representation properties.

Among these representation properties we find some that are used to store texts (such as the text that would be shown to identify the entity or the property, or even the text that would be shown depending on the status of the element), some others represents the object that would be shown (a button, slider, etc.) and others describe the picture that represents the element (the URL to the image, its size, position, etc.).

In Figure 2 an example of the interface, generated for a lamp (a) and a dimmable lamp (b) is shown. The lamp represented in (a) has the capability “isSwitchable”, so it

[2] http://qrcode.kaywa.com/
[3] http://code.google.com/p/zxing/

has a button to turn it on and off. On image (b), there is a dimmable lamp control interface. This lamp is composed of a reading light and a halogen one, so it has two capabilities: "isSwitchable" and "isDimmable", which are represented by a button and a slider, respectively.

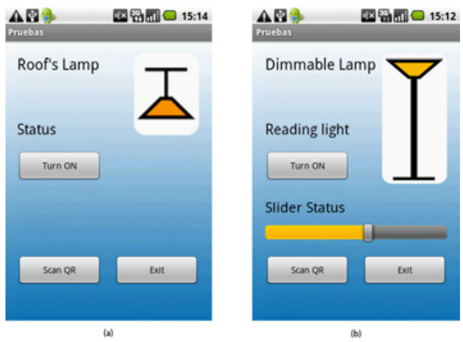

Figure 2. Screenshots of control interfaces for a lamp (a) and for a dimmable lamp (b)

3. Evaluation

To test our proposal, we have evaluated the system with external users. Two classical measurements to evaluate a system are effectiveness, number of correct attempts of a set, and efficacy, the length of time that a user takes to complete a task . Individual questionnaires are also used to evaluate subjective measurements, such as satisfaction or usefulness.

3.1. Effectiveness

The effectiveness of the system has been measured as the number of correct readings of QR Codes. We also wanted to study the relation between the size of the QR Code, the distance and the practical and aesthetic the QR Code was.

The QR Codes used for the experiment were generated and printed in four sizes: XL, L, M and S, 8 cm., 5.4 cm., 4 cm. and 3.4 cm. wide respectively. Users had to capture each of them at three different distances: 100 cm., 50 cm. and 25 cm. To do that, each user was given five attempts to capture each QR Code from each distance.

From the data collected we have prepared the graphics showed in Figure 5. They represent the histogram of the correct attempts for every size of the QR Codes for each distance.

If we study each graphic, we conclude that the probability of doing a correct capture depends narrowly on the distance. The biggest QR Codes (XL) present higher rates in every distance than smaller ones (M and S), which only obtain good results (over 90% hit rate) for distances lower than 25 cm.

As we said before, we wanted to find a well-balanced relation between the QR Code size, the distance, the correct attempt rate and the suitability. We understand the suitability as, on the one hand, aesthetics. The users' satisfaction could be reduced if she had to mark all the elements in her environment with big tags. She could feel some kind of intromission from the system in her everyday environment. On the other hand, we consider suitability as the ratio between the size of the QR Code and the element. Some devices (even the smallest ones) should be tagged with marks adapted to their own size.

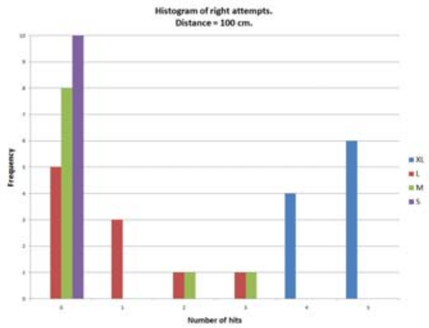

Figure 3. Histogram of correct attempts.

Distance = 100 cm.

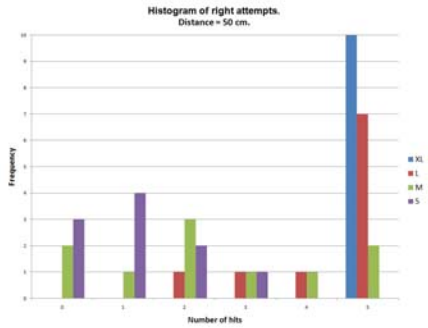

Figure 4. Histogram of correct attempts.

Distance = 50 cm.

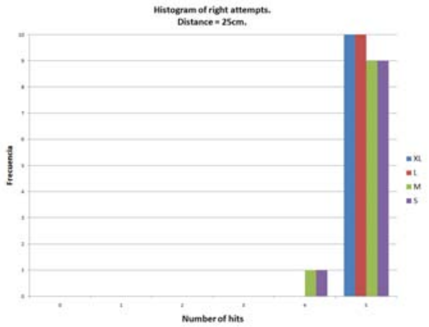

Figure 5. Histogram of correct attempts.

Distance = 25 cm.

Although XL QR Codes offer higher hit rates on each distance; the aesthetics and suitability are not always the best. It is big enough to cause a visual impact on the decoration of the environment or to block up the element that it represents. The L size presents a good relation between size, distance, hit rate and suitability. It is big enough to offer good correct captures rates (better for distances under 50 cm.) And it is suitable for most devices in an environment. S size should only be used in case of necessity.

3.2. Efficacy

Efficacy was measured in terms of the length of time the user took to perform these three tasks:

- Turning on roof's light
- Changing the value of a dimmable light (higher intensity)

- Visualize the photography of a person (people are also defined in the blackboard and, among their properties, they have a photography that can be represented)

The average time needed to perform each task is represented in the graphic shown in Figure 6.

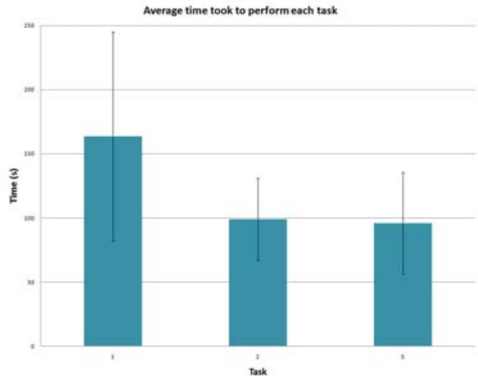

Figure 6. Average time took to perform each task

The variation in time that a task requires may be caused by the familiarity that the user could have with this kind of devices (smart phones) or applications. The reason why the great majority of the testers dedicated longer to perform the first task may be motivated by: on the one hand, it was the first time they used the application and they were not familiar with it. On the other hand, every time a user started the application, she had to authenticate in the system.

Tasks 2 and 3 took shorter time, in general terms. This reduction in the performing time may be motivated by the fact that the users already knew the functionality of the application and also because instead of restarting the application to scan another QR Code, they selected the correct option to scan another code without restarting.

3.3. Subjective Questionnaire

After completing the attempts of capturing QR Codes and performing the three tasks, each user was asked to fill a subjective questionnaire where she had to answer a set of questions related to four topics: usefulness, ease of use, ease of learning and satisfaction. Each of these sections consisted of a set of questions (4 to 10) to be evaluated between 1 (fully disagreement) and 5 (fully agreement).

With the data collected from the questionnaires, we elaborated the graphic in Figure 7. The graphic represents the average score given by users to each topic of the questionnaire.

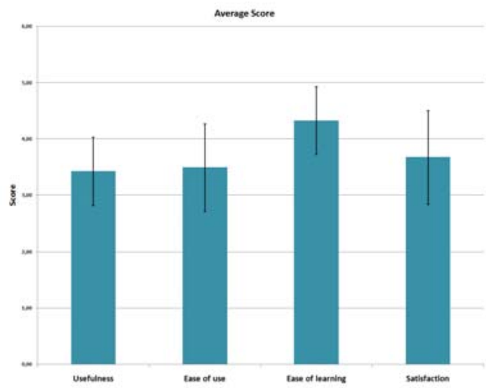

Figure 7. Average score for each questionnaire' topic

- **Usefulness**: The majority of users considered the system useful. Some questions of this section were asked to evaluate the feelings about the system, in terms of time saving and productivity improvement. Users felt a bit disappointed in these aspects, probably because of the fails in the scanning tasks or the necessity of rescanning the QR Codes each time.
- **Ease of use**: Testers considered the system to not require a big effort to be understood, although a small tutorial or an instruction manual was appreciated. Besides some users seemed to be disappointed with the ability of the system to recover from a fail situation.
- **Ease of learning**: Users considered easy to learn how to operate with the system. Once they were explained how the system worked, and because it was an almost identical task to control different devices, users did not have any problem to perform the tasks proposed.
- **Satisfaction**: Users presented a high satisfaction rate with the system, feeling comfortable while using it.

4. Conclusions

In this article a system that generates ubiquitous user interfaces for Ambient Intelligence environments has been presented. These interfaces are intended to be used in mobile devices.

Mobile devices (PDAs, smart phones, etc.) are actually popular among the society and, thanks to the continuous improvements in their technological capabilities, they are open to new market and research niches. On the other hand, Android seems to fit perfectly with this research because it is an open mobile platform that is becoming more popular everyday and, the devices that run this mobile operating system are equipped with advanced hardware parts, such as high resolution cameras, GPS, multi-touch screens, etc.

Our job is based on the "ubiquitous remote controller" metaphor. Thus, our idea is that a user could be able to control any element of the environment just by pointing at it with her smart phone. This system identifies the element that is being pointed at by the use of QR Codes. The interface is generated automatically from the information extracted from the QR Code and from the supporting architecture (the blackboard). In that sense, the blackboard has the capability of storing generic interface descriptions of the entities that compose it.

Users, unfamiliar with the application, have evaluated the software prototype. The results we have obtained are acceptable, but they could be improved in future versions. One of the main topics that should be improved is efficacy (the system should be faster, error–free, etc.), so the satisfaction of the user might be improved as well.

Finally, we consider that a system that generates ubiquitous user interfaces in an agile way and with an effective management seems to be suitable for controlling ambient intelligence environments. It will be able to be used anywhere, at any moment and by users without previous knowledge of each element of the environment.

5. Future work

The system is still in a first step of development and further research has to be carried out. From the data collected in the questionnaires we conclude that we have to effort on managing error sources and giving more feedback to the user: progress bars, informative messages, alerts, etc. but in a subtle way. It would be desirable to allow the user to configure the message detail and the amount of them that are prompted.

On improving the effectiveness of the system we propose to add a cache memory of interfaces. This way, once the QR Code is scanned and decoded, this cache memory could store the interface so, if a user decides to interact with that element at any other moment, she would not have to rescan the code. This mechanism seems to be useful in home environments where users often control the same devices.

We think there are several other possibilities to indicate the system the element we want to control, such as selecting from a list, writing the name of the element or by means of oral interaction. These new possibilities could improve the user experience although they should not distance from the metaphor of the ubiquitous controller.

On the other hand, thanks to the fact that the system requires the user to authenticate, it could be possible to know who is using the interface at any moment. Therefore, we could adapt the interface to her preferences or special needs. The same QR Code and element should produce different interfaces depending on user needs and characteristics. This is a major subject of research that will have to be addressed in the short future.

Acknowledgements

This work was partially funded by projects eMadrid (Comunidad de Madrid, S2009/TIC-1650), Vesta (Ministerio de Industria, Turismo y Comercio, TSI-020100-2009-828) and HADA (Ministerio de Ciencia y Educación, TIN 2007-64718)

References

[1] Weiser, M. 1994. The world is not a desktop. ACM Interactions, 1(1), 1994.

[2] Shafer, S., et al. 2001. Interaction Issues in Context – Aware Intelligent Environments. Human – Computer Interaction, 16, 2001.

[3] Haya, P. A., Montoro, G., y Alamán, X., A Prototype of a Context -- Based architecture for Intelligent Home Environments. Conference on Cooperative Information Systems (CoopIS), 2004.
[4] Haya, P., y Montoro, G., A spoken interface based on the contextual modelling of smart homes. HCI related papers of Interacción, 2004.
[5] Gómez, J., Montoro, G., y Haya, P., iFaces: Adaptative user interfaces for Ambient Intelligence. IADIS International Conference on Interfaces and Human Computer Interaction, 2008.
[6] García-Herranz, M., Haya, P., Esquivel, A., Montoro, G., y Alamán, X., Easing the smart home: Semiautomatic adaptation in perceptive environments. Journal of Universal Computer Science 14(9), 2008.
[7] Rekimoto, J., Ayatsuka, Y., CyberCode: designing augmented reality environments with visual tags. Proceedings of DARE 2000 on Designing augmented reality environments, 2000.
[8] Wagner, D., y Schmalstieg, D., First steps towards handheld augmented reality. Proceedings of the 7th IEEE International Symposium on Wearable Computers, 2003.
[9] Billinghurst, M., Kato, H., Weghorst, S., y Furness, T., A mixed reality 3d conferencing application. Human Interface Technology Laboratory, University of Washington, Technical Report R-99-1, 1999.
[10] Wagner, D., Pintaric, T., Ledermann, F., y Schmalstieg, D., Towards massively multi-user augmented reality on handheld devices. Third International Conference on Pervasive Computing, 2005.
[11] Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavári, Z., Encarna\c cao, L., Gervautz, M., y Purgathofer, W., The studierstube augmented reality project Presence: Teleoperators and Virtual Environments. MIT Press 11, 2002.
[12] Schmidt, D., The ADAPTIVE Communication Environment: An object-oriented network programming toolkit for developing communication software Concurrency: Practice and Experience. 1993
[13] Siltanen, S., y Hyväkkä, J., Implementing a natural user interface for camera phones using visual tags. Proceedings of the 7th Australasian User interface conference-Volume 50, 2006.
[14] Al-Khalifa, H., Utilizing QR Code and Mobile Phones for Blinds and Visually Impaired People. Proceedings of the 11th international conference on Computers Helping People with Special Needs, 2008.
[15] Gómez, J., Montoro, G., Haya, P., García-Herranz, M., y Alamán, X., Easing the integration and communication in Ambient Intelligence. International Journal of Ambient Computing and Intelligence 1(3), 2009.
[16] Kato, H., Tan, K., Pervasive 2D barcodes for camera phone applications. IEEE Pervasive Computing, IEEE Computer Society, 2007.