

# Testing the Usability of a Platform for Rapid Development of Mobile Context-Aware Applications

Valentim Realinho<sup>1,2</sup>, A. Eduardo Dias<sup>2</sup> and Teresa Romão<sup>2</sup>

<sup>1</sup> Escola Superior de Tecnologia e Gestão/Instituto Politécnico de Portalegre  
Lugar da Abadessa, Apartado 148,  
7300-901 Portalegre, Portugal  
valentim.realinho@estgp.pt

<sup>2</sup> CITI, DI-Faculdade de Ciências e Tecnologia/Universidade Nova de Lisboa  
Quinta da Torre  
2829-516 Caparica, Portugal  
{tir, aed}@di.fct.unl.pt

**Abstract.** In this paper, we present the usability evaluation of IVO (Integrated Virtual Operator), a platform that supports the rapid development of context-aware applications by users with no programming skills. Using only the tools provided by the platform (IVO Builder and IVO Outlook), users can define temporal and spatial conditions and associate them with workflows of activities available within the platform. This way, whenever the defined conditions occur, the user's smartphone will immediately produce the intended behavior, with no need for user intervention. The applications developed using IVO can easily be made available to other users through a distributed web platform. Moreover, an android client was developed, to run the IVO-developed applications, allowing the smartphone to act as the ubiquitous interaction device. The evaluation of the platform was performed through usability tests at both the end-user level (android client) and the developer-user level (builder tools).

**Keywords:** Ubiquitous Computing, Context-Awareness, Rapid Application Development, Usability Evaluation, Interaction Design, Mobility.

## 1 Introduction

In Mark Weiser's vision, we should be able to live with computers and not only to interact with them [1]. The basic idea behind the ubiquitous computing paradigm is that computing moves off the desktop into the environment, operating in the periphery of our attention. Greenfield describes this interaction paradigm as "information processing dissolving in behavior" [2].

In essence, this paradigm uses small devices, with robust computing power and network capabilities, seamlessly integrated into the environment, involving this heterogeneity of devices with a high degree of interoperability. Strides in nanotechnology are enabling more and more features that are added to small devices allowing a greater awareness of the dynamic world they are part of.

Pister introduced the term “*smart dust*”, referring to tiny sensors, no larger than grains of rice that would monitor everything, acting like electronic nerve endings for the planet [3]. In this world the user is no longer part of the system (since the interaction between the system and the user is reduced) and becomes the supervisor of the system, acting only when strictly necessary [4]. However, these devices are not part of the actual ubiquitous computing paradigm yet, while the mobile phone is the first truly pervasive computer [5].

The mobile phone is always with people, helping them to keep in touch with each other and manage everyday tasks. In this project, the approach was to use smartphones as ubiquitous interaction devices, thus enabling a widespread use of the system. Smartphones provide a large set of possibilities for sensing, communication and user interface, and there is a great market demand for these types of devices [6], making them ideal for the development of ubiquitous systems.

Although using smartphones limits context-awareness possibilities to the use of the sensors available in this kind of devices, they also facilitate the widespread use of these type of applications with no or easy setup. However, the former drawback tends to disappear since smartphones are increasingly including a larger number of sensors. The use of Near Field Communications (NFC) [7] in mobile devices opens a new spectrum for context-aware applications, which was possible only through hardware and additional infrastructure. The latest version of Android already supports NFC and iPhone 5 is expected to feature NFC support.

An important research topic addressed in ubiquitous computing is context-awareness, where applications can dynamically adapt to changes in the user activities and environments [8]. The context is a particular situation for each user, and it can include, among others, location, activity being performed, time and nearby people or equipments. Context-aware computing involves sensing those situations to provide adequate information and services to the user.

Pioneering work on ubiquitous computing research, like [1, 9-11], addresses the topic of context-awareness and has demonstrated the potential of context-aware applications. Some research has focused on developing frameworks, toolkits and infrastructures to support programmers in building context-aware applications. Other projects use end-user programming techniques, which empower end-users to prototyping context-aware applications [8, 12, 13]. However, they still lack the end-user support for the creation of complete applications using the currently available smartphones as ubiquitous interaction devices.

To address this problem, we have developed IVO, a platform that enables end-users to build and deploy context-aware applications without the need to write any programming code. IVO provides a completely visual programming environment that allows end-users to define a number of context conditions and workflows of activities, which are later triggered when the user is in the presence of those contexts. The use of a control-flow model facilitates the visualization of how activities are interconnected, taking advantage of the humans’ innate spatial reasoning skills.

Three main building tools are currently available, allowing end-users with no programming skills to create mobile context-aware applications: IVO Builder, a web application, used for the definition of location contexts and respective workflows of activities; and IVO Outlook, for temporal and proximity contexts, composed of two *add-ons*, for Outlook Calendar and Outlook Contacts. The set of available workflow

activities provides a wide range of capabilities, while keeping the system accessible to both novice and expert end-users. An Android smartphone client was also developed to provide the workflow engine that uses the stored model to activate the required workflow when, and where, the activation conditions are met by the user.

## 2 Related Work

This section summarizes how IVO relates to context-aware computing. We divide this analysis in three categories: Middleware and Frameworks, End-user Prototyping and Authoring Tools.

**Middleware and Frameworks.** Projects like Context Toolkit [11], SOLAR [14], Context Fabric [15], JCAF [16] and Placelab [17] are targeted at programmers. They provide the infrastructure to enable them to create context-aware applications, but they do not provide any support for end-users. Of these, the Context Toolkit is one of the most widely referenced projects. It provides a software interface to physical sensors, separating the acquisition and representation of context from the delivery and reaction to context. The authors proposed context widgets as components to encapsulate the sensing details, providing context abstraction.

**End-user Prototyping.** *a CAPpella* [8], Topiary [12], Papier-Mâché [18] and iCAP [13] use end-user programming techniques in order to empower end-users to prototype context-aware applications. *a CAPpella* is a programming by demonstration context-aware prototyping environment intended for end-users, using four main components: a recording system, an event detection, a user interface and a machine learning system. Topiary allows the "demonstration" of scenarios representing local contexts, and then it uses these scenarios to create storyboards that describe interaction sequences, which can run on mobile devices. Papier-Mâché is a toolkit for building tangible interfaces using computer vision, electronic tags, and barcodes. iCap, is a system that allows end-users to describe a context and associate an action with it. It offers an interface that allows the creation of *if-then rules* and then associates actions to it.

**Authoring Tools.** We include in this analysis, projects that provide authoring tools, because we found some similarities with IVO, although intended for specific domains. They enable the rapid authoring of applications by domain specialists. Hull et al. [19], propose a framework that allows the rapid authoring of mediascapes. The applications developed can be downloaded and executed on handheld computers augmented with sensors. UbiCicero is a multi-device, location-aware guide supporting museum visits, which also provides the possibility of enriching the museum visits through individual or collaborative games [20]. It allows museum curators to easily customize the contents for their museum guides through an authoring environment.

Besides these projects, we also found some similarities in standalone applications for the Android platform, as Tasker [21] and Locale [22], which seek to automate

tasks based on context information. These applications allow users define in the smartphone, a set of tasks that are triggered when a defined condition occurs, while IVO allows the creation of more complete applications. We allow the easy definition of contextual information associated with each situation and the activities to be performed in a workflow logic. IVO also allows the sharing of applications through a distributed web platform and the location and temporal contexts of IVO are richer.

IVO differs from these projects in many ways. IVO allows the creation of fully functional applications with a wide range of uses. Our focus was on the reaction to context and not on the acquisition. Thus it provides a workflow environment for the actions which allows developing more complex applications. IVO's user interface is web-based which facilitates its use globally. Finally, IVO uses widely available smartphones as the ubiquitous interaction device, therefore allowing for a widespread use of the applications created by the end-users.

### **3 The Solution**

To address the lack of support for end-users to build context-aware applications based on current smartphones, our approach was to provide end-users with a toolset that should allow them to: (i) easily define spatial, temporal and proximity conditions, and associate them with workflows that are triggered when the defined conditions occur; (ii) easily make the created applications available to other users through a distributed web platform, making the system available worldwide; (iii) create applications that can be used for various purposes from leisure to a more professional or business use; and (iv) run applications in currently available smartphones like Android, iPhone and Windows Mobile devices.

To address the first requirement, we developed a web application, called IVO Builder, which allows the creation of applications based on location. Outlook Calendar and Outlook Contacts were chosen for the user definition of temporal and proximity contexts, therefore allowing users to make use of their previous experience with these kind of popular applications. For the definition of the temporal conditions, we have created an Outlook Calendar *add-on* that allows the incorporation of workflows in events, which are executed "*when the event starts*" or "*when the event ends*". For proximity contexts, we have created an *add-on* for Outlook Contacts, which facilitates the discovery of Bluetooth devices in the proximity and associates the selected device to the contact. We named both *add-ons* as IVO Outlook. The synchronization tools provided by the device manufacturer guarantee the synchronization of IVO Outlook with the smartphone.

The second requirement was solved with a REST web-service which provides the services that the ubiquitous device (the smartphone) can use for the synchronization of applications that the user has access to. When creating an application the user becomes its owner and decides who can use it. The developed applications can be set to private (only the owner can use them), public (available for all users) or shared (only the selected users can use them).

The set of activities available in workflow design enables users to produce applications targeting different uses, e.g. personal or business, which addresses the third requirement. Possible activities range from changing the phone profile, integration with social networks or GPS navigation, to forms and quizzes, web-services from third parties, text-to-speech or voice recognition. The complete set of possible activities is explained later in this paper.

Finally, an Android client was developed providing the first fully functional client of the system running on one of the most widespread mobile platforms. Furthermore, this Android client allowed for a user evaluation of the builder tools.

We can identify three core features in IVO's architecture: a rule based system, an event-driven workflow and a visual and control-flow programming environment.

**Rule based system.** We base the system on *if-then rules*, meaning things like "if I ..." or "when I ..." am in a particular situation, perform this action. Most context-aware applications can be described in that way. The study conducted by Dey et al. [13], on the way users think about context-aware applications, concluded that most users express themselves in terms of *if-then rules*. The conditions of *if-then rules* are evaluated through the Java Expression Language (JEXL) [23], using context information sensed by the smartphone. The use of JEXL allows a great expressiveness of the conditions with the use of common operators like >, >=, <, <=, !=, AND and OR.

**Event-driven workflow.** We use an event-driven workflow paradigm, in which workflows are started by events that result from the evaluation of the *if-then rule* engine. Workflows determine the program flow, expressed as a sequence of activities, which can include alternative or conditional paths and cycles.

**Visual and Control-flow Programming.** Control-flow programming is used to build workflows because it makes it easy for the user to understand how things connect, unlike with imperative programming, which focuses on how things happen. We use visual programming environments to create the complete workflow and the rest of the elements that make up the applications, since these type of environments have proven to be effective in taking advantage of the user's spatial reasoning skills [24].

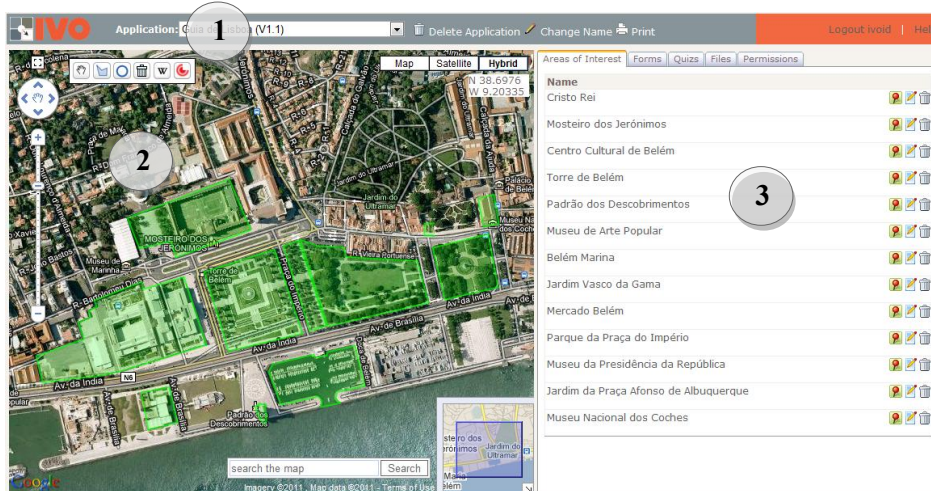
## 4 The IVO Interfaces

Fig. 1 illustrates the main screen of the IVO Builder, loaded with an application for a Tourist Guide of Lisbon. The system gives access only to registered users, with each user developing his/her own applications in their area of the server. Each of them can, however, make the developed applications available to other users.

On the top area of the main screen (zone 1 in Fig. 1) there are options that allow the application management, including loading, creating, changing, deleting and printing. The left side of the screen (zone 2 in Fig. 1) is reserved for the geographical representation, which is overlaid with the IVO areas of interest, represented in green.

The toolbar in this area gives access to basic features for creating areas of interest, namely: (i) manual creation of an area by defining the polygon that defines it; (ii) creation of an area represented by a circle; (iii) import areas from geo-referenced articles on Wikipedia; (iv) import areas from existing areas in Wikimapia. In addition to the toolbar, there are the usual controls to zoom and change the map view, satellite or hybrid, and a search box that allows users to quickly find a place through Google search APIs. The right side of the main screen (zone 3 in Fig. 1) is reserved for the representation of information on the application itself, such as areas of interest, forms, quizzes, files (including images, audio, video and scripts) and permissions to other users. The forms built with the IVO Builder are compatible with XForms [25].

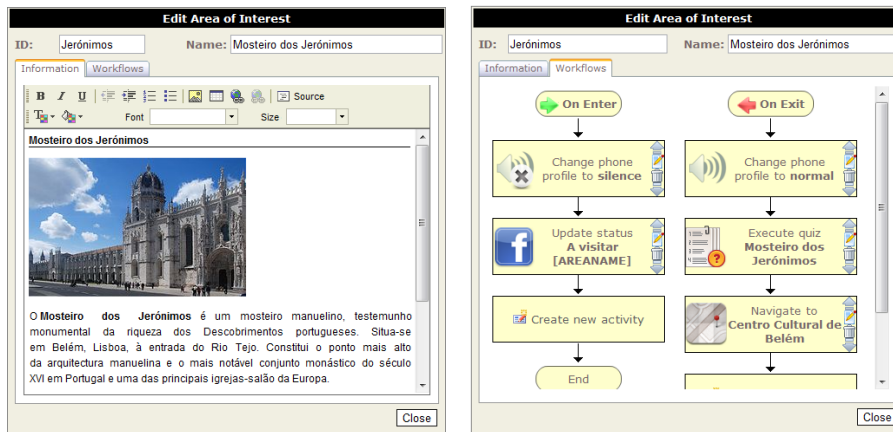
For more advanced users, we have a workflow activity for executing scripts. The scripting languages available are those natively supported by the smartphone platform in use. For example, on Android we use the Scripting Layer for Android (SL4A) [26] which allows executing scripts coded on Python, Perl, JRuby, Lua, BeanShell, JavaScript, Tcl, and shell. These scripts have access to many of the Android APIs, which makes it possible to create activities that are not viable otherwise, when using only the IVO workflow activities.



**Fig. 1.** Main screen of IVO Builder, loaded with an application for Tourist Guide. 1 – Application manager menu with options for loading, creating, changing, deleting and printing; 2 – The map helps the user to create location contexts. The areas of interest of IVO are highlighted in green. 3 – This area contains IVO application components, like the representation of the location contexts, forms, quizzes and files including images, audio, video and scripts.

Once created the areas of interest, the user can associate them with contextual information (Fig. 2a), using an HTML editor, or linking them to a website that will provide this information. On the same screen, the "Workflows" tab allows the creation of the entry workflows (Fig. 2b), which are executed when the user enters the area, and of the exit workflows, that are executed when the user exits the area. The example below shows (a) the information displayed to the users when entering the area of

Mosteiro dos Jerónimos, Lisbon, Portugal (Wikipedia page) and (b) the entry workflow marked with green arrows (pointing right) as well as the exit workflow marked with red arrows (pointing left). In this case, when the user enters the specified area, the phone profile changes to silence mode and the user's status on facebook is updated with the defined message. When the user exits the area, the phone profile changes to normal mode and IVO launches a quiz about the point of interest the user has just visited. After completing the quiz, the user is guided to the next point by the navigator.



























**Fig. 2.** Screens for the edition of areas of interest, where the user can define (a) the information about the point of interest and (b) the entry and exit workflows.

Workflows are flows of activities controlled by the contextual data gathered by the smartphone as defined by the users. Presently, we only support sequential workflows, but we have already started the development for the support of more complex workflows with conditional or alternative paths and cycles, represented through an *event-driven finite state machine*. The activities that can currently be performed at each step of the workflow are illustrated in Table 1. The creation of more activities is a task that involves programming an abstract class extension.

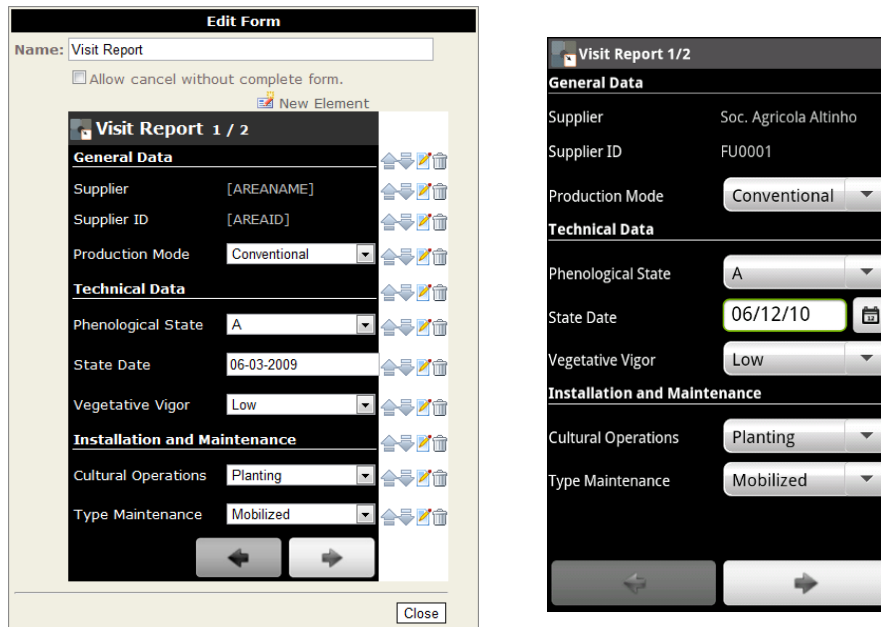
The IVO Builder includes tools for creating quizzes and forms, which can be used as workflow activities.

A quiz can contain multiple questions with a maximum of four possible answers each, and it is possible to configure how they are displayed on the IVO client. For example, a minimum score (in percentage) can be adjusted to make the application immediately jump to the next workflow activity when the user reaches that score. The quiz can also be spoken to the user using the text-to-speech engine, when available on the client side.

**Table 1.** Activities currently available for the workflow construction.

Icon	Activity	Icon	Activity	Icon	Activity
	Change phone profile		Update status on facebook		Execute external web-service
	Make a call		Open a web page		Read IVO code (QR-code)
	Send a SMS		Enter a value		Detect nearby people or equipment
	Send a email		Speech the text entered (text-to-speech)		Run script
	Don't show information window		Show alert message		Run external application
	Play audio file		Show toast message		Navigate to a place
	Play video file		Complete form		Take a photo
	Post message on twitter		Execute quiz		Voice recognition

The forms built with the IVO Builder are compatible with XForms [25]. The developed Android IVO Client includes an XForms processor that is able to run that type of forms. Fig. 3a) shows the screen to create a form used in an application for the agricultural sector while Fig. 3b) shows the same form running on the IVO Client. Data supplied by users while completing the form are stored on the server, allowing them to view these data as predefined reports or export them for external use.



**Fig. 3.** a) The design of an IVO Form with a form loaded. b) The same form running on the Android IVO Client.



As mentioned before, the temporal conditions are defined in Outlook Calendar. This IVO Outlook *add-on* adds a form region at the bottom of the appointment window containing the interface that enables the users to define workflows with the same activities that are available in IVO Builder and pointed above. Fig. 4 illustrates an Outlook Appointment with a workflow that starts "when the event starts" and a workflow that starts "when the event ends". In this case, the activities defined by the user are: change phone profile and update facebook status. Users can associate the event to a location context created in IVO Builder or to the address field defined in user Contacts. This way, the system can handle more complex conditions like "at this instant in time and when I'm at this location with these people" do these actions.

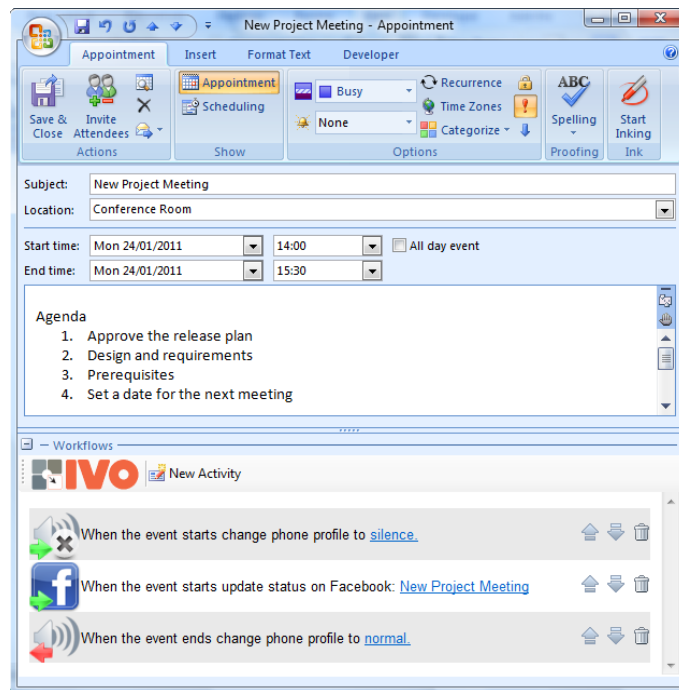


Fig. 4. Outlook Appointment with IVO workflows.

## 5 Usability Evaluation

Usability tests played an important role in this work, because they helped us to observe different users while interacting with IVO tools. We conducted usability testing to determine and understand possible usability issues in IVO. IVO Builder and IVO Outlook were tested separately.

The IVO Builder has been tested by two groups of volunteers. A group of eleven graduate students in Computer Science with two females among them and a group of six Design college students being one of them a female. With these two groups we intended to assess the extent to which the area of expertise of the users would

influence the results. The subjects for this experiment ranged in age from 20 to 43 years old with a mean age of 23.8.

IVO Outlook has been tested by a group of five graduate students in computer science (all male) and a female PhD student in Computer Science. All the participants had their first contact with the system during the test and used it under similar conditions. The subjects for this experiment ranged in age from 21 to 33 years old with a mean age of 24.5.

The test sessions were conducted individually for each user by two investigators, who played the roles of facilitator and observer. Before starting using the system, users were informed about the objectives of the test, and received a list describing the tasks they should perform (no reference on how to do it was made). The users were encouraged to "think aloud" and tell what was going through their minds while using the system. This way, users could freely explore the interface and we could clearly identify the usability issues. During the tests, the observer took notes on the user's behavior and verbalizations while performing the tasks, and the duration of each test was measured.

## 5.1 Test Scenarios

The test scenarios used in this evaluation were designed in order to allow for the testing of most of the features available within the system. The details of the test scenarios are described below.

**Tourist Guide.** In this scenario, the user was challenged to create an application using the IVO Builder to guide tourists through points of interest in the city of Lisbon. It was intended that when tourists arrive at points of interest, contextual information about them is shown using Wikipedia as a source. The phone should also be placed in silent mode and a wall post should be posted on facebook, informing the user's friends that she/he is visiting the spot. At the end of the visit to each location, the phone's profile should be switched to normal mode and the GPS Navigator would guide the tourist to the next point of interest. At certain points, the tourist would be challenged to answer some questions (quiz) about the location just visited. At the end, the user had to respond to an electronic questionnaire to evaluate the roadmap experience.

**Using IVO Outlook.** In this scenario the user was asked to use IVO Outlook to create contacts and associate them with a Bluetooth device (the smartphone) in order to allow activities of type "nearby people", and to create appointments to which the user then adds workflows. The user must create a new appointment, and create workflows that will run at the beginning and at the end of the appointment. When the appointment starts, the phone should be placed in silent mode and a message should be posted on facebook. The phone must also speak a message notifying the user for the start of the appointment. When the appointment ends, the phone should be switched to normal mode.

## 5.2 Questionnaire

After performing the usability tests, users were asked to answer a questionnaire which captured personal data, experimental feedback, as well as users' suggestion and comments. Personal data included the participant's age, gender, education level, familiarity with new technologies and frequency of use of Internet, computer, mobile phone, and game console. Experimental feedback was evaluated through four different sections of the questionnaire. The first section included three statements regarding the application ease of learning (Table 2). Users indicated their level of agreement with each statement by circling a value on a 5-point Likert-type scale, with a response of 1 (one) meaning "strongly disagree" and a response of 5 (five) meaning "strongly agree".

The second section included seven statements concerning the application ease of use criteria and one question on how easy it was to perform the tasks available in the application (Table 2). To classify the seven statements users followed the same procedure used for the first section as described above. To answer the last question, a scale from 1 (one) meaning "very difficult" to 5 (five) meaning "very easy" was used.

**Table 2.** Statements that composed the experimental feedback part of the questionnaire of IVO Builder. IVO Outlook questionnaire differs only in the tasks of question 11 which are: a) Create an IVO Contact; b) Create an IVO appointment and c) View IVO contacts and appointments.

<b>Statement</b>
Ease of learning
Q1 It is easy to learn to use the application
Q2 I quickly became able to use the application
Q3 I easily remember how to use the application
Ease of use
Q4 It is easy to use
Q5 It requires few steps to accomplish what is intended
Q6 It can be easily used to build other applications, scenarios or situations
Q7 I can use it without written instructions
Q8 I don't notice any inconsistencies as I use the interface
Q9 It was easy to recover from an unexpected situation
Q10 It reacted quickly to user actions
Q11 How easy it was to:
a) Create application
b) Create an area of interest by importing data from Wikipedia
c) Create an area of interest by importing data from Wikimapia
d) Manually create an area of interest (polygon)
e) Change the polygon that defines an area of interest
f) Build a quiz
g) Build a form
h) Define workflows

In the third section users were asked to indicate the three activities they considered most useful and the three ones they considered less useful. An open question allowed users to suggest other activities they consider useful to be available in IVO builder.

The fourth section included one question based on the Microsoft "Product Reaction Cards". These aimed at capturing the user's feelings when using the system, since they facilitate the measuring of intangible aspects of the user's experience [27]. Users were asked to choose the words that best described their experience while using the system from a list of 24 words, consisting of about 60% of words considered positive and 40% considered negative. Users could choose how many words they wanted.

Finally, the questionnaire also included an open question in order to gather comments and recommendations regarding future developments of features and to obtain a more general evaluation of the system.

## 6 Results

The users' answers to the questions presented in Table 2 were analyzed and the results are shown and discussed below. We examined the average scores of the users' responses to see if there were general trends in their opinions (strong feelings one way or the other showing up as mean scores closer to 1 or 5). We used the standard deviation of the mean score to determine how broad the consensus about the issue was. Fig. 5 and 6 summarize the results for the questions presented in Table 2 concerning the IVO Builder and the IVO Outlook respectively.

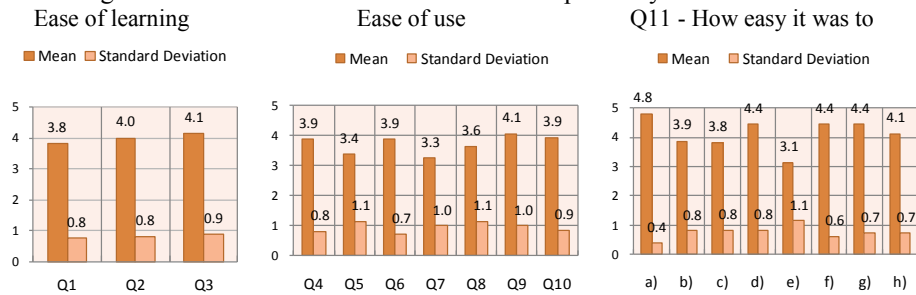


Fig. 5. IVO Builder results

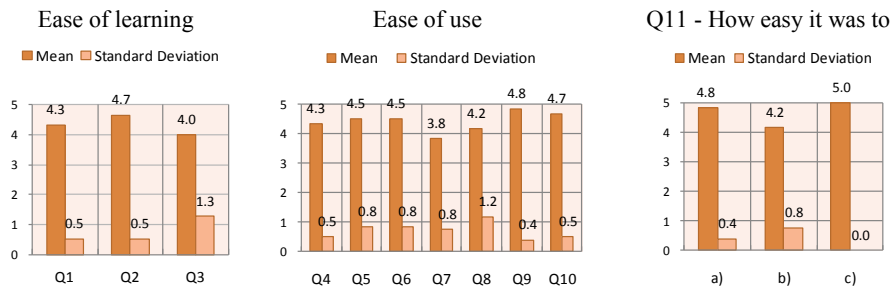


Fig. 6. IVO Outlook results

In general, the user's feedback was quite positive. Question 7, regarding the ability to use the application without written instructions, received the lowest score (although positive) for both applications.

Overall, users rated IVO Outlook better than IVO Builder, proving that it was easier to create temporal contexts than location contexts for the testing scenario. The biggest difficulty in IVO Builder was found in the task to change the polygon that defines an area of interest (Q11e). As we have already anticipated this difficulty, we had prepared written instructions that were provided to the users only in case they were struggling. After providing the instructions, users were able to accomplish the task without any difficulty.

**IVO Builder.** All participants were able to accomplish all tasks, although there were some occasional difficulties, especially when they were asked to change the polygon defining an area of interest. The time to complete the test ranged from 28 minutes to 1 hour and 5 minutes. The two groups that performed the tests, did not show a significant difference, which means that, in this case, the users' different areas of knowledge did not influence the results. However, some curious behaviors were observed in group 2 (design group), typical of desktop applications users, like the use of CTRL-Z to undo, BACK key to delete and drag and drop in some operations.

Most participants stated that it's easy to remember how to use the tool (statement 3) and they quickly became skillful with it (statement 2). Although very positive, the users' feelings about the statement "It is easy to learn to use the application" (statement1) was not so strong.

The majority of the participants stated that it was easy to use the IVO Builder (statement 4) and that it was easy to recover from an unexpected situation (statement 9). Although positive, the participants' opinions concerning the number of steps to accomplish what is intended (statement 5), the application use without written instructions (statement 7) and the inconsistencies found when using the interface (statement 8) show lower scores and higher standard deviation revealing weak consensus between users. The majority regarding the application's "ease of use" (Q4-Q10) results are positive, but the answers to four of the questions (revealed a standard deviation higher or equal than 1, revealing a lower consensus in users' opinions.

Question 11, assessed how easy users found each of the proposed eight tasks (described in Table 2). According to the results displayed in Fig. 5, the interactions were generally deemed easy to perform. The only exception was the task to change the polygon that defines an area of interest (statement 11e) which users found not so easy to perform. Their opinions were very heterogeneous and they conform to the users' observed behavior.

**IVO Outlook.** All participants were able to accomplish all tasks, without any difficulty, in a period between 9 and 16 minutes. The user's feedback was quite positive and consensual. Most participants agreed with all the statements. The lowest score was found in the statement about the application use without written instructions (statement 7). We considered it a very subjective matter, since the participants have not used any written instructions and we did not notice any difficulty when they were performing the tasks.

In the third section of the questionnaire participants were asked to indicate the three activities they considered most useful and the three they considered less useful. Analyzing the results, we conclude that it seems to be a pattern in the users' responses regarding the more useful activities. Users tend to refer those used in the test: change phone status (20 users), navigate to (12 users) and update status on facebook (7 users). The activities considered less useful were "voice recognition" (7 users) and "run a script" (4 users). We believe that, since participants haven't used these activities during the test, they may not have realized how they can use for.

From the analysis of the fourth part of the questionnaire (Fig. 7) we concluded that 90% of the participants held positive feelings when classifying their experience using the system. The most selected word was "useful", followed by the word "simple" and "pleasant". 44% of the participants considered the system "professional" and "satisfactory", and 38% "innovative". Only two users reported negative feelings and the words used were "confused", "complex" and "boring".

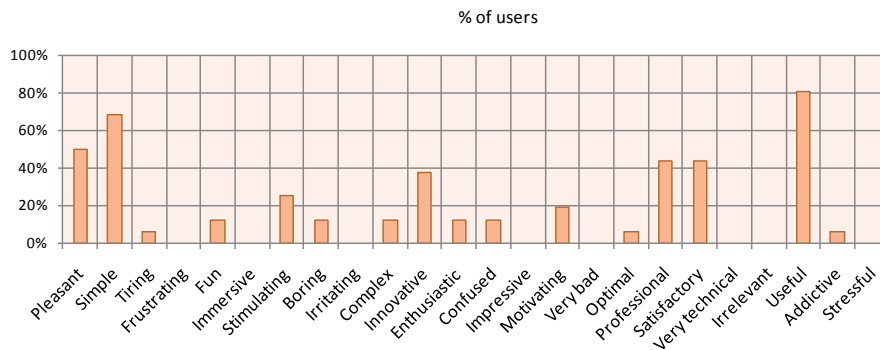


Fig. 7. User's feelings when using the system

Overall, the evaluation results were very positive, the users expressed very encouraging feelings about the tested tools and they declared themselves very willing to use it as they found it very useful.

## 7 Conclusions and Future Work

We have presented IVO, a platform for building context-aware applications by end-users without the need to write any code, using smartphones as the ubiquitous interaction device. It opens up the space of context-aware computing to end-users, allowing them to build fully functional context-aware applications.

The phone is considered the first truly ubiquitous device. Most top end smartphones come with a handful of sensors and it is expected that in future new sensors will be added, extending the spectrum of use for context-aware applications.

The event-driven workflow approach enables the development of a wide range of applications. An application consists of *if-then rules* and workflows that are triggered when the conditions occur. We plan to support more sophisticated workflows with

alternative paths, cycles and also new activities, in order to make the system even more expressive. We also intend to improve the building tools with modules for testing and debugging.

The tests allowed us to conclude that even first-time users could build context-aware applications through the platform and revealed some minor interface issues, already solved. They also showed that visual programming environment facilitates the rapid development of such applications.

IVO is now starting to be used in a project in the area of Persuasive Computing aiming at enhancing people's behavior towards the environment. In this context the developed applications should induce positive changes in the way users deal with daily activities such as recycling and energy or water consumption. As future work we also want to develop IVO clients for other mobile platforms, such as the iPhone, therefore reaching a larger number of users.

## 8 Acknowledgments

This work was funded in part by ILUSTRATOWN, by Agência de Inovação (ADI) under contract 70/2007/33B/00216/00178 and by FCT – Fundação para a Ciência e Tecnologia, Portugal, in the scope of project DEAP (PTDC/AAC-AMB/104834/2008).

## 9 References

1. Weiser, M.: The Computer for the 21st Century. *Scientific American* 265, 94-104 (1991).
2. Greenfield, A.: *Everyware: The Dawning Age of Ubiquitous Computing*. New Riders (2006).
3. Brett, W., Matt, L., Brian, L., Kristofer, S.J.P.: Smart Dust: Communicating with a Cubic-Millimeter Computer. *Computer* 34, 44-51 (2001).
4. Want, R., Borriello, G., Pering, T., Farkas, K.I.: Disappearing Hardware. *IEEE Pervasive Computing* 1, 36-47 (2002).
5. Rafael, B., Jan, B., Michael, R., Jennifer, G.S.: The Smart Phone: A Ubiquitous Input Device. *IEEE Pervasive Computing* 5, 70 (2006).
6. International Data Corporation (IDC), <http://www.idc.com/about/viewpressrelease.jsp?containerId=prUS22560610>.
7. Finkenzeller, K.: *RFID Handbook : Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons (2010).
8. Dey, A.K., Hamid, R., Beckmann, C., Li, I., Hsu, D.: a CAPpella: programming by demonstration of context-aware applications. *Proceedings of the SIGCHI conference on Human factors in computing systems*. pp. 33-40. ACM, Vienna, Austria (2004).
9. Schilit, W.N.: *A System Architecture for Context-Aware Mobile Computing*, PhD thesis, Columbia University, (1995).
10. Harter, A., Hopper, A., Steggle, P., Ward, A., Webster, P.: The Anatomy of a Context-Aware Application. *Wireless Networks* 8, 187-197 (2002).
11. Dey, A.K., Salber, D., Abowd, G.D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction* 16, 97-166 (2001).

12. Li, Y., Hong, J.I., Landay, J.A.: Topiary: a tool for prototyping location-enhanced applications. Proceedings of the 17th annual ACM symposium on User interface software and technology. pp. 217-226. ACM, Santa Fe, NM, USA (2004).
13. Dey, A., Sohn, T., Streng, S., Kodama, J.: iCAP: Interactive Prototyping of Context-Aware Applications. In: Fishkin, K., Schiele, B., Nixon, P., and Quigley, A. (eds.) Pervasive Computing, vol. 3968. pp. 254-271. Springer Berlin / Heidelberg (2006).
14. Chen, G., Kotz, D.: Solar: An open platform for context-aware mobile applications. Proceedings of the First International Conference on Pervasive Computing (Pervasive 2002). pp. 41-47 (2002).
15. Hong, J.I.: The context fabric: an infrastructure for context-aware computing. In CHI '02 extended abstracts on Human factors in computing systems. pp. 554-555. ACM, Minneapolis, Minnesota, USA (2002).
16. Bardram, J.E.: The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications. In: Gellersen, H.W., Want, R., and Schmidt, A. (eds.) Pervasive Computing, vol. 3468. pp. 98-115. Springer Berlin / Heidelberg (2005).
17. Hightower, J., LaMarca, A., Smith, I.: Practical Lessons from Place Lab. IEEE Pervasive Computing 5, 32-39 (2006).
18. Klemmer, S.R., Li, J., Lin, J., Landay, J.A.: Papier-Mâché : Toolkit Support for Tangible Input. Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 399-406. ACM, Vienna, Austria (2004).
19. Hull, R., Clayton, B., Melamed, T.: Rapid authoring of mediascapes. In: Davies, N., Mynatt, E., and Siio, I. (eds.) UbiComp 2004: Ubiquitous Computing. pp. 125-142. Springer Berlin / Heidelberg (2004).
20. Ghiani, G., Paternò, F., Santoro, C., Spano, L.D.: UbiCicero: A location-aware, multi-device museum guide. Interact. Comput. 21, 288-303 (2009).
21. Tasker for Android, <http://tasker.dinglich.net/>.
22. Locale for Android, <http://www.twofortyfouram.com/>.
23. Java Expression Language (JEXL), <http://commons.apache.org/jexl>.
24. Shu, N.C.: Visual programming: perspectives and approaches. IBM Systems Journal 28, 525-547 (1989).
25. The Forms Working Group, <http://www.w3.org/Markup/Forms>.
26. Scripting Layer for Android (SL4A), <http://code.google.com/p/android-scripting>.
27. Benedek, J., Miner, T.: Measuring Desirability: New methods for evaluating desirability in a usability lab setting. Proceedings of UPA (Usability Professional' Association). Orlando, USA (2002).