

Interaction and End-User Programming with a Context-Aware Mobile Application

Jonna Häkkinen¹, Panu Korpipää², Sami Ronkainen¹, and Urpo Tuomela¹

¹ Nokia Multimedia,
Yrttipellontie 6, 90230 Oulu, Finland
firstname.lastname@nokia.com

² VTT Electronics,
Kaitoväylä 1, P.O. Box 1100, FI-90571 Oulu, Finland
panu.korpipaa@vtt.fi

Abstract. In this paper we present the user interface design and evaluation of a tool for customizing mobile phone applications with context-aware features. The tool enables the user to program a set of context-action rules, defining the behavior of the device when a selected context is recognized and/or some other user-defined conditions are met. The tool user interface design is described starting from an early paper prototype and its evaluation, leading to a functional software implementation in a mobile phone. Finally, the usability evaluation of the functional prototype, and other relevant findings from the user test, are presented.

1 Introduction

Context-awareness has recently been intensively studied in mobile computing. Context-awareness concerns proactive actions, such as user interface adaptation or evoking appropriate application functions in certain external conditions. The literature gives several studies on location-aware applications, which have been demonstrated in, for instance, the Lancaster tourist guide *GUIDE*, the Aware Campus at Cornell University [3], or in offering location and presence information in Active Campus [2]. Sensor-based context recognition systems have been presented in [5] and [11].

Despite the extensive research in the field, several problems remain in developing usable context-aware applications. Uncertainties in context recognition or ambiguity may lead to erroneous conclusions of current context [4], and objective and explicit definition of context attributes and their values is often problematic. Moreover, using contextual information for automated actions has led to concern about whether context-awareness will take the control away from the end user [1]. Enabling end-user programming of context-aware features in mobile phone applications solves the problem of user control, but brings other challenges; end-user programming is an unknown concept for most mobile phone users. Moreover, mobile phones are commonly used because of their easy accessibility and ability to perform multi-tasking – but it is as yet unclear how large a percentage of users would like to configure their phones, even if significant personalization and usability efficiency advantages were offered. Hence end-user programming is a challenging field, and so far has been very little

studied among mobile handheld devices. Automated capture has been investigated in [6], and its usage linked to end-user programming in [16].

In this paper we present the user interface and usability evaluation of a mobile phone application, Context Studio, the concept of which was originally introduced by Mäntyjärvi et al. [12] and further developed by Korpipää et al. [8, 9]. Context Studio is a tool that enables the user to define context-action rules in order to activate mobile phone functions when the rule conditions are fulfilled. The user-centric design practices were emphasized in the user interface design process. The main contribution of this paper is the description of the user interface design and results of the user test, which was arranged to evaluate the final design with a functional prototype in a mobile phone. The tool was developed on a Symbian Series 60 platform following its style guides.

The paper is organized as follows. First, the background and experience from the Context Studio development are shared. Then the tool functionality and user interface are introduced. This is followed by the evaluation of the functional tool against the criteria based on standard usability guidelines. Finally, a discussion and conclusions, with plans for future work, are presented.

2 Context Studio

Mobile phones are used in many different kinds of situations, i.e. in numerous different contexts, where the priorities of the device functions and most applications vary according to the situation. Furthermore, mobile phones are personal devices usually constantly carried by the user, and thus adapting the device behavior to the specific user is a relevant goal. The ways of using a mobile phone differ significantly among individuals. The aim is to provide more flexibility of use for various user groups.

2.1 Development Background

As mentioned earlier, selecting incorrect contextual information sources may lead to inappropriate actions being performed by the device. It is thus important to select relevant information sources in the design phase, or enable customization of the sources. In sensor-based interaction the recognition of correct patterns from the measured signals is crucial for success.

User control and visibility of system status are two important issues in Nielsen's usability heuristics [13], which is a commonly used guideline in usability evaluation. In the design of the tool, one of the main goals was to provide the sense of user control, which has been one of the usability concerns with context-aware applications [1]. In addition to this, the visibility of system status is a design challenge, as the current state of the device (and user) context may be hard to define explicitly as a union of all available context types [12].

The approach to these problems was to let the user decide which context types they wanted to apply to each situation [8], instead of having to define all context types that describe the situation. The tool provided a selection of contextual triggers and application actions, which were presented in a hierarchical folder structure for a scalable representation and to aid the user navigation.

The experiences from the previous studies [12, 8] were utilized in developing the tool user interface. According to the previous experiences, we decided to leave out complex rule structures, including a set of Boolean operators, as they were found to complicate the rule definition for most users. On the other hand, the idea of defining trigger-action rules had been perceived as understandable and intuitive. It was found that users generally prefer making simple rules. However, by not using Boolean operators, setting expressive rules would become problematic. As a compromise, the Boolean operator AND was included as the default operator between multiple triggers in the same rule, while OR was made available by allowing the user to define parallel rules for the same set of triggers.

Context Studio was designed and implemented for the Nokia Series 60 platform phones, which sets certain restrictions, such as screen size and style. The Nokia Series 60 user interface style guide was utilized for defining the look-and-feel and general interaction style for the tool.

An iterative HCI design was employed during the development process. The first evaluation of the Context Studio user interface was done with paper prototypes early in the development process, and feedback was used to iterate the user interface design before the software implementation was started. In the next section, the Context Studio user interface is described as it was in the final evaluation.

2.2 User Interface

Context Studio is a mobile phone tool that enables the user to define context-action rules. The condition (context) part of the rule is called a trigger for clarity. The rules take a simple predicative format:

*When the trigger condition is fulfilled,
the action is executed.*

For instance:

*When the environment sound intensity is loud,
set phone ringing tone volume up.*

The user can also combine multiple triggers with the AND operator, for instance:

*When the location is home and the phone battery is charging,
save new images into an image album.*

The triggers can include implicit (context) inputs - such as location, environment sound and device activity - or explicit input actions - such as gestures, if gesture recognition is supported - or RFID-based commands. Figure 1 shows an example screenshot from the Context Studio rule view.

In Figure 1 the screenshot of the Series 60 mobile phone display shows the basic elements of a user-defined rule. The first row presents the name of the rule, which is automatically generated from the *action* and *trigger*. Although the automatically generated name appears when the user selects the elements for the rule, (s)he can also edit the name. If the user defines the name before setting the action and trigger(s), the name is not automatically generated.

User-defined rule for a gesture-triggered phone action

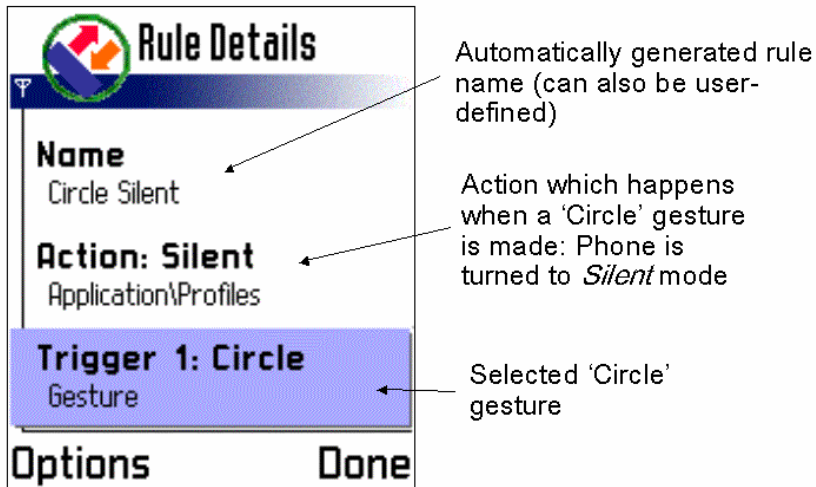


Fig. 1. Rule view in Context Studio

Below the rule name are the *action* and *trigger* fields. The user selects the triggers and actions from a hierarchical folder structure, where the items are named and grouped according to their meaning and resemblance. Below the attribute values, which are marked with bold text in Figure 1, the path to the value, describing the attribute type, is shown. This provides the user with additional information on the attribute value, and helps the user to remember the overall folder structure. An example of the folder structure appearance is presented in Figure 2. Folders describe the attribute type, and can contain subfolders, which the user can navigate to find the desired attribute value. For example, the Environment folder in Figure 2 contains the subfolders Temperature, Light, and Humidity. The Light folder contains the subfolders Intensity, Type, and SourceFrequency. The last folder in the path contains the attribute values to select from. Hence the folder hierarchy enables a scalable representation of a large number of triggers and actions to choose from, instead of, e.g., displaying a very long list of values.

To enable the setting of more complex and sophisticated rules, the user can include more than one trigger in the rule. This is done by selecting 'Add Trigger' from the 'Options' menu, which results in a 'Trigger 2' field appearing below the 'Trigger 1' field. The defined triggers are connected with the Boolean operator AND.

In order to keep the rules understandable and relatively simple, it was decided that a single rule could contain only one *action*. However, to provide a quick way of enhancing the automated action with the same contextual triggers, we wanted to provide a shortcut for creating the rules, and included a rule copy function. The user can also activate, deactivate, delete, send, and edit rules. Active rules are marked with an icon in the main rule list, and active attributes are marked with an icon in the attribute value list. The color of the icon indicates the state of the corresponding attribute. Furthermore, a set of phone joystick and keypad controls were implemented for each user interface view to allow flexible use of the tool.



Fig. 2. The folder structure in the *triggers* view. The folders describe the attribute type and can contain subfolders, which the user can navigate to find the desired attribute value.

2.3 Enabling the Functional System

A functional prototype system was implemented to enable the user to create rules with the tool and after that use the features defined in the rule. After the user has created the desired context-action rule, the context framework handles the background monitoring of contexts and the triggering of actions according to the rules [9]. Context management is based on the blackboard-based context framework for mobile devices. For example, after the user has created the rule shown in Figure 1, (s)he can change the profile of the phone to silent at any time by making a circle-shaped gesture with the phone.

The folder hierarchies for *actions* and *triggers* are automatically generated based on a context ontology model, described in detail in [8]. The ontology vocabulary hierarchy is transformed into a folder-file model representation in the UI. The ontology, with an enhanced vocabulary model, offers a scalable information representation, and easy navigation of context and action information in the UI, and enables straightforward updating of the UI elements according to changes in the vocabularies - i.e., without programming any changes to the tool itself. Furthermore, the ontology supports the utilization of a formal rule model. Rules are represented as Context Exchange Protocol (CEP) XML scripts, which can be executed by an inference engine [10].

3 Evaluation

Before evaluating the functional prototype, the user interface of the tool was tested with two iterations of paper prototypes (the latter is briefly described in section 3.1) during the development process, and improved accordingly [12, 8]. Nielsen's usability

ity heuristics [13] were applied during the UI design process. The iterative user testing during the development is considered valuable as it reduces the need for corrections after the implementation.

3.1 Early Evaluation with Paper Prototypes

A user test was conducted with paper prototyping early in the development process. Eight subjects (4 male and 4 female), aged 20–39, from different fields of study or work participated in the test. The subjects had no previous experience of context-aware systems, and none were involved with the mobile phone industry. Each test session was conducted at controlled premises and lasted about 1.5–2 hours. During the test the subjects conducted given tasks with the paper prototype user interface according to given usage scenarios.

The results of the test were used to redesign the user interface before the software implementation. The main modifications concerned the automated name generation for the rules and the disposition of some elements in the screen layouts. The results also affected the order of the *action* and *trigger(s)* elements, see Figure 1, since it was discovered that it was preferable to have both elements of the rule in the same view. Thus the *Action* field was placed above the *Trigger 1* as the user might select several triggers, even though placing the triggers first would have been more logical.

3.2 Final Evaluation – The Evaluation Criteria

According to the standard on Human-Centred Design Processes for Interactive Systems, usability is defined as the ‘extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use’ [14]. The criteria for the evaluation were adopted from an ergonomic requirements standard’s usability guidelines [15], and were complemented with case-specific measures. The goal of the user study was to verify the selected fundamental customisation approach and the tool UI, and to gain possible feedback for further development. Moreover, initial user feedback was collected from the novel customisable smartphone interaction modalities. The evaluation criteria were the following.

Easy to learn: Formally, learnability is the resources expended before a specified level of proficiency in terms of effectiveness and efficiency is attained, or the speed at which proficiency increases over time. Learnability was assessed with two measurements: the ability to operate the tool without instructions on the first try, and the number of tasks that were required before reaching the ability to use the tool without errors. Folder searching was not considered an error. The measurements were based on usage monitoring, and oral and written feedback.

Effectiveness: According to the ISO 9241 standard on usability, effectiveness is the accuracy or completeness with which users achieve specified goals. The main measurement here was the ability of the user to complete a given task successfully.

Efficiency: Efficiency is the expenditure of physical and mental resources with which users achieve specified goals. It concerns time and the physical and cognitive effort that needs to be spent to successfully complete the given task. In this case, physical

and cognitive effort can be measured as, e.g., the number of errors, re-doing basic actions, or consulting the manual or asking advice.

Satisfaction: Satisfaction refers to the level of comfort when using the tool, and is measured as the user's attitudes (positive or negative) towards the tool. The measurement was based on the subjective written feedback given by the test participants after the tests.

3.3 Test Set-Up

The evaluation of the application was done with user tests by ten participants (4 male, 6 female), none of whom worked in the mobile phone industry. The test was carried out with a Series 60 Nokia mobile phone, see Figure 3. The tests were video recorded with a screen camera. The users were encouraged to think aloud during the test, and their comments were recorded. After completing the tasks, the users completed a questionnaire that charted their opinions on the tool and the modalities, and their ideas for development.



Fig. 3. The tool user interface in the mobile phone used in the user tests

The test consisted of five scenarios, based on which the users were asked to use the briefly introduced customisation tool without further advice. The scenarios only implicitly referred to making rules, such as scenario 1:

You often call your best friend, Anna, and you want to be able to make the call quickly and without watching the phone, as you often call her while riding a bike. The phone recognizes a 'circle' movement. Now you want to determine that when you make a circle with the phone, it starts calling Anna.

Table 1 presents the contexts and actions that were used in the rules. In each table cell the attribute type (folder) is the first element and the attribute value is the second

element. Optional rule elements are in brackets. The scenarios in Table 1 can be read as, for instance, scenario 1: When the user makes a Gesture with value Circle, perform an Application function Call with value Anna.

The context and action vocabularies, transformed into a UI directory hierarchy, consisted of over a hundred name-value pairs, which the users navigated in order to select the elements of the rule.

Table 1. The user customised five features. In scenarios 1,3, and 5 the user could perform the actual interaction with the smartphone after the customization.

Scenario	Context trigger(s)	Action
1	Gesture Circle	Application\Call Anna
2	Gesture Square (& Location Home)	ExternalDevice\TV On
3	Device\Movement\Swing SwingRight	Application\Profiles Silent
4	Location OuluCenter & Environment\Sound\Intensity Loud (& Environment\Light\Intensity Dark)	Application\Profiles Loud
5	Connections\RFTag Jonna	Call Jonna

In addition to the scenarios the users were asked to perform separate tasks, such as training personal gestures. The aim was to collect feedback on how the users responded to the possibility of customising personal triggers, in addition to the rules.

3.4 Results

Results According to the Criteria

In the following, the results of the test are discussed separately for each criterion presented in each section.

Easy to learn. The idea of constructing context-action rules was understood by 9/10 users at the beginning of the first task on a general level: ‘one must design what happens when certain conditions are fulfilled’. During the first task, 4/10 users needed clarification of the terms *action* and *trigger*. All ten participants could complete the tasks without help after performing the first task. Use of the tool is easy to learn.

Effectiveness. All participants were able to complete all given tasks. All participants were able to complete the task given in scenario 1, although one participant required help with the fundamental idea of rule building. For scenarios 2-5, all ten participants could perform the given tasks. In scenario 4, the task was given so that it involved the users setting several trigger options in more detail. Here the results showed some variations in the selected triggers. This result is consistent with earlier research on user perceptions and context-awareness - i.e., the users tend to have a subjective definition for what is meant by context [12]. Moreover, another finding was that the participants were not eager to define long settings, even if they understood that this would result in more precise rules. Thus, to summarize, the usage of the tool is effective.

Efficiency. The number of errors was small after the first task, and the users did not have to ask advice or redo basic actions. However, since the content of the context trigger and action directories was unknown to the users beforehand, they initially had to perform some searches to find the required rule elements. This behaviour was obvious as the participants used the application for the first time. The choice of ontology concepts is crucial for the UI navigation efficiency. The usage of the tool is quite efficient.

Satisfaction. In the written feedback, 8/10 participants said that they would benefit from the application. When they were asked to rate the general usability of the application on a scale from 1 (worst) to 5 (best), the average was 3.7 with a standard deviation of 0.8. Median of the rating was 4, minimum 2 and maximum 5. The usage of the tool is quite satisfactory.

User Feedback

The users were asked which of the features in the test they liked the most. All ten subjects answered the question, and four of them gave more than one answer. Gesture control was preferred in six answers, implicit interaction based on sensed contexts in five, and three most liked controlling external devices with the phone. RF tag-based physical selection was not mentioned as a most-liked feature, which may be due to the currently required extra hardware and the difficulty in quickly perceiving the possibilities of an interaction concept completely unknown to users. In this sense, gestures have the advantage that they are innately natural to human communication. Concerning implicit control, the users had different opinions about the subjective contexts, such as Environment\Sound\Intensity Loud. This suggests that subjective contexts should either be defined by the users themselves, or omitted, or at least the description of the meaning of the subjective context value should be accessible. The result confirms our earlier experience [12].

For gesture control, the users were asked which method of gesture recognition they preferred; the user marks the gesture by pressing a button, or continuous recognition without marking the gesture with a button. The participants unanimously perceived user-initiated activation of gesture recognition as better than continuous. The main reason for this was a concern about unintended gesture commands. Two female participants were worried about false commands when the phone was carried in a handbag. Moreover, training personal gestures was appreciated because it was assumed to improve the recognition accuracy and enable the use of intuitive and natural gestures. This is in line with our earlier results [7]. When asked to comment freely, one user proposed training of location context, e.g. *'I would [like to] go to the bar and press a button, and the device would remember the place'* (user #10). Customised locations could then be used as triggers for rules.

As a summary of the evaluation results, the user study verified the feasibility of the customisation approach. The implemented tool was found to be usable, and targets for further development were acquired. Moreover, initial user feedback was collected about the new modalities for phone control. The users most liked the idea of freely trainable gestures, controlling external devices with the phone, and implicit context-based interaction.

4 Discussion

A few general observations based on the user tests are in order. The subjects were more willing to construct short and simple rules than more complex ones, even though they were advised that they could define several triggers. Configuring the phone settings was generally perceived as an additional hassle, and people tended to set rules with just one trigger. Perceptions of the usefulness of Context Studio differed depending on the given scenarios. The perceived usefulness of the tool depended on whether the scenario appealed to the personal needs and usage experiences of each test subject. The ability to personalize the phone and provide shortcuts to the actions corresponding to their usage style and behavior was generally appreciated.

Thus the users had preferences on which triggers and actions they liked. The underlying software design enables tailoring the available triggers and actions - i.e., it is possible that the user may delete unnecessary triggers and actions from the folder structure. This has not yet been implemented, but is considered future work.

An alternative to explicit rule setting would be to model contexts based on examples, which is feasible when an example contains a single, chosen type of context. When one example contains multiple types of contexts, the programming-by-demonstration approach may lead to functionality that the user did not intend to have, if the user cannot control exactly which contexts are relevant for the intended action. Hence the explicit rule definition approach gives the user better control over creating the intended device functionality.

Although the user tests give a good indication of a successful interaction design, the true value of the tool and the ways in which the users would use it can only be revealed by observing long-term usage, which is relevant further work. However, the evaluation shows that the resulting user interface is both understandable and usable. Furthermore, the tool was evaluated with a functional prototype developed on a widely used mobile platform by applying a novel software framework and information representation methods, thereby demonstrating the feasibility and potential for utilization by large user groups.

5 Conclusions

This paper presented the user interface design and user evaluation of a tool for end-user programming of context-aware applications developed for a Series 60 mobile phone platform. The tool enables the user to define context-action rules that cause the mobile phone to perform actions when the rule conditions are fulfilled. The design process and rationale for the interaction design were discussed, and the resulting mobile device user interface was presented. The tool user interface and the fundamental customization approach were evaluated with user tests. The results were analyzed against criteria based on standard usability guidelines. The results indicate that the tool is easy to learn and efficient to use.

The study provides a good basis for further studies. Future work related to interaction design includes user-initiated capture of context attributes, more flexible personalization of the contextual information to better support the user's individual needs, and a large-scale user deployment to study the utility of the tool in daily life.

References

1. Barkhuus, L., and Dey, A.: Is Context-Awareness Taking Control Away from the User? Three Levels of Interactivity Examined. *Proceedings of Ubicomp (2003)* 159-166
2. Barkhuus, L., and Dourish, P.: Everyday Encounters with Context-Aware Computing in a Campus Environment. *Proceedings of Ubicomp (2004)* 232-249
3. Burrell, J., Gay, G. K., Kubo, K., Farina, N.: Context-Aware Computing: A Test Case. *Proceedings of Ubicomp (2002)* 1-15
4. Erickson, T.: Some Problems with the Notion of Context-Aware Computing. *Communications of the ACM, Vol. 5, No. 2 (2002)* 102-104
5. Gellersen, H.W., Schmidt, A., Beigl, M.: Multi-Sensor Context-Awareness in Mobile Devices and Smart Artefacts. *Mobile Networks and Applications Vol. 7 (2002)* 341-351
6. Hayes, G. R., Kientz, J. A., Truong, K. N., White, D. R., Abowd, G. D., Pering, T.: Designing Capture Applications to Support the Education of Children with Autism. *Proceedings of Ubicomp (2004)* 161-178
7. Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., Di Marca, S.: Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing special issue on Multimodal Interaction with Mobile and Wearable Devices, Springer-Verlag, In Press*
8. Korpipää, P., Häkkinen, J., Kela, J., Ronkainen, S. & Känsälä, I.: Utilising context ontology in mobile device application personalisation. *Proceedings of International Conference on Mobile and Ubiquitous Multimedia, ACM (2004)* 133-140
9. Korpipää, P., Malm, E., Salminen, I., Rantakokko, T., Kyllönen, V. & Känsälä, I.: Context management for end-user development of context-aware applications. *Proceedings of International Conference on Mobile Data Management MDM'05 (2005), In Press*
10. Lakkala, H.: Context Exchange Protocol Specification, Context Script Specification. (2003), Available: <http://www.mupe.net>
11. Michahelles, F., and Samulowitz, M.: Smart CAPs for Smart Its – Context Detection for Mobile Users. *Personal and Ubiquitous Computing Journal, Vol. 6, Springer-Verlag London Ltd. (2002)* 269-275
12. Mäntyjärvi, J., Tuomela, U., Häkkinen, J., Känsälä, I.: Context-Studio - Tool for Personalizing Context-Aware Applications in Mobile Terminals. In *Proceedings of Australasian Computer Human Interaction Conference OZCHI'03 (2003)* 64-73
13. Nielsen, J. & Mack, R.L.: *Usability Inspection Methods*. Canada: John Wiley & Sons, Inc (1994)
14. Standard: ISO 13407 (3.3)
15. Standard: ISO 9241-11:1998. Ergonomic requirements for office work with visual display terminals (VTDs). Part 11: Guidance on usability. (1998)
16. Truong, K. N., Huang, E. M., Abowd, G. D.: CAMP: A Magnetic Poetry Interface for End-User Programming of Capture Applications for the Home. *Proceedings of Ubicomp (2004)* 143-160