Multimodal Interaction on Mobile Phones: Development and Evaluation Using ACICARE

Marcos Serrano[1], Laurence Nigay[1],
Rachel Demumieux[2], Jérôme Descos[2], Patrick Losquin[2],

[1] CLIPS-IMAG
38000, Grenoble, France
{marcos.serrano, laurence.nigay}@imag.fr

[2] France Télécom Division R&D
22307, Lannion, France
{rachel.demumieux, jerome.descos, patrick.losquin}@francetelecom.com

ABSTRACT

The development and the evaluation of multimodal interactive systems on mobile phones remains a difficult task. In this paper we address this problem by describing a component-based approach, called ACICARE, for developing and evaluating multimodal interfaces on mobile phones. ACICARE is dedicated to the overall iterative design process of mobile multimodal interfaces, which consists of cycles of designing, prototyping and evaluation. ACICARE is based on two complementary tools that are combined: ICARE and ACIDU. ICARE is a component-based platform for rapidly developing multimodal interfaces. We adapted the ICARE components to run on mobile phones and we connected them to ACIDU, a probe that gathers customer's usage on mobile phones. By reusing and assembling components, ACICARE enables the rapid development of multimodal interfaces as well as the automatic capture of multimodal usage for in-field evaluations. We illustrate ACICARE using our contact manager system, a multimodal system running on the SPV c500 mobile phone.

Categories and Subject Descriptors

H.5.2 **[Information Interfaces and Presentation]**: User Interfaces – *Input devices and strategies, Interaction styles, Prototyping, User interface management systems (UIMS), Evaluation/methodology*; D.2.2 **[Software Engineering]**: Design Tools and Techniques – *User interfaces*

General Terms

Algorithms, Human Factors

Keywords

Multimodal Interface, Mobile Device, Software Component, Mobile Multimodal Logging, Field trial.

1. INTRODUCTION

Mobile devices, such as Pocket PCs and Smart-Phones, are becoming increasingly powerful. This evolution is seriously compromised by the limited interaction capabilities e.g. the restrictions of small displays and keypads.

Multimodality seems to be a good solution to enhance usability on mobile devices by allowing the user and the system to choose the pure or combined modalities according to the user preferences, the task at hand and the variable social and physical contexts of use. Experimental studies on mobile devices confirmed the flexibility of multimodal interaction. For example, based on a woz platform we experimentally studied multimodal interaction on a PDA for reading electronic mails [18] and we highlighted the effective use of multimodality, inter and intra individual differences, appearance of preferential tendencies and change of modalities in the dysfunction situations. Moreover in [13], they experimentally observed the use of alternative modalities on mobile phones.

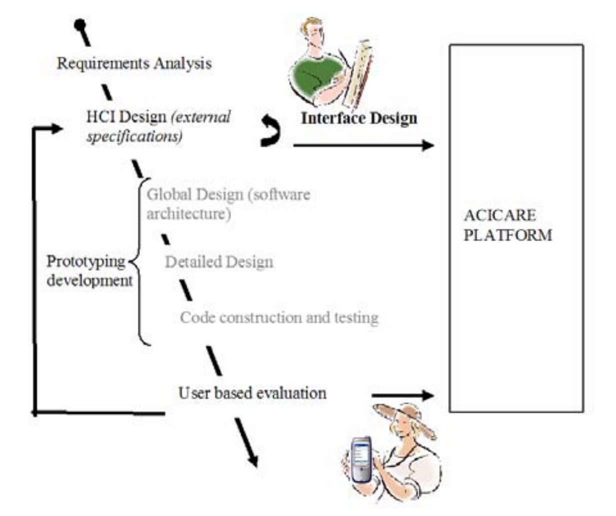


Figure 1. Iterative design process using the ACICARE platform.

Nevertheless, as pointed out in [13], to gain understanding and user acceptance of multimodality on mobile devices, it is crucial to study tools that support a truly iterative design process which consists of cycles of designing prototyping/development and evaluation. In this paper we address this issue by presenting a platform, namely ACICARE, that is dedicated to the overall iterative design process of mobile multimodal interfaces as shown in Figure 1. Indeed ACICARE

- allows rapid and easy development of multimodal interfaces on mobile devices, and,
- provides automatic usage capture that will be used for the evaluation of the multimodal interface.

MobileHCI'06, September 12–15, 2006, Espoo, Finland.
Copyright 2006 ACM 1-58113-000-0/00/0004…$5.00.

The structure of the paper is as follows: the next section examines related work. We then give an overview of the example that has been implemented with ACICARE, a contact manager application running on a mobile phone. In the sections following we present the ACICARE platform, composed of ICARE and ACIDU, detailing how those tools connect together. We conclude with the case study application specification and results.

2. RELATED WORK

The implementation and evaluation of multimodal applications on mobile devices is not straightforward.

With regards to development issues, the mobile devices have characteristics that make implementation very different from what can be done on desktop PCs. Differences such as context of use, input mechanisms (drag-and-drop does not exist) and display (non support of layered windows) imply that there are few software tools which specifically address those devices. In this context, software tools for mobile multimodal interaction are even fewer. Motorola, Opera Software and IBM created a multimodal mark-up language standard called XHTML+Voice (X+V) that provides a way of creating multimodal Web applications [17]. IBM’s Multimodal Tools Project is an Eclipsed-based tool whose purpose is to support and speed up the building of multimodal (X+V) Web applications. All these tools/standards focus on one form of multimodality: the functional equivalence of modalities by adding speech to GUI. Kirusa, a developer and licensor of multimodal wireless platforms, has developed the Kirusa Multimodal Solution (KMS), a product for the deployment of multimodal solutions by mobile carriers and enterprises. KMS is based on markup languages, such as VoiceXML [16] or WML. KMS is going one step further towards multimodal interaction by providing combined use of speech with pointing gestures.

All the above mentioned standards and tools address the challenge of multimodal web applications and are based on markup languages that define platform independent vocabularies and toolkits. Such approaches belong to the more general issue of plastic User Interfaces (UI) [15] and address the challenge that UIs on mobile devices must accommodate the capabilities of different access devices with various interaction resources. As explained in [14], in addition to marked languages that are adopted by mass market, other approaches (more in the research community) for addressing the whole issue of plastic UIs adhere to a model-based user interface development. Such tools are based on several models of the interaction at different levels of abstraction [6]. One problem of such an approach is that the designer has less control on the final UI: as explained in [10] "For every user interface, it is important to control the low-level pragmatics of how the interactions look and feel, …". To solve this known problem, mixed approaches that is both top-down (Abstract UI to Concrete UI) and bottom-up are defined as in [7] and [14].

Our approach belongs to the model-based approach but focuses on multimodal input interaction and does not address the problem of plastic UIs. As opposed to the majority of existing development tools and approaches for mobile devices, our goal is not the adaptation of the UI to various mobile platforms but the study of multimodal interaction on mobile platforms. Nevertheless we are not opposed to these approaches and in the future ACICARE can be a useful tool as part of a model-based development environment for plastic UIs for defining the model of multimodal interaction. Moreover since ACICARE is a tool dedicated to the overall iterative design process (design, development and evaluation), it will help in gaining understanding of multimodality on mobile devices towards the establishment of guidelines for mobile multimodal interaction that can be incorporated in a model-based development environment.

In addition to the rapid development of mobile multimodal interaction, ACICARE supports automatic usage capture for evaluations in the field. While some researchers argue that such evaluations in the field are not of value [9], in [1] they recently showed the usefulness of both laboratory and field trials. Evaluations on mobile phones are difficult to carry out as mobile phones are used in mobility, in different contexts and in a private way. Mobile video methods that record the user test limit the mobility of the user, a small camera being fixed on the mobile devices or worn by the user as in [12]. Complementary to video methods, ACICARE implements a logging system that enables on-board multimodal data collection in a realistic mobile environment. As opposed to the MATCH multimodal logger [8] allowing high-fidelity records that can be replayed, ACICARE only focuses on the capture of input multimodal interaction. Our goal is to provide a tool that enables us to quickly explore several input multimodal interaction on mobile devices based on continual user testing in realistic situations.

Before presenting the ACICARE platform that combines ICARE for developing input multimodal interaction on mobile phones and ACIDU for capturing multimodal usage data, we present an example for illustrating our approach whose main features are presented in the next section.

3. ILLUSTRATIVE EXAMPLE: A MOBILE PHONE CONTACT MANAGER

Using ACICARE we built a multimodal application on the mobile phone SPV c500. The application consists of a contact manager for creating new contacts. The application allows the use of several interaction modalities.

In the present version of the contact manager on the SPV c500, the creation of new contacts uses only the keyboard device. For creating a new contact, the phone user must fill a form composed of several fields, as shown in Figure 2: first name, last name, work phone, etc. We decompose this “fill form” task into a series of “fill field” tasks. In turn we decompose each “fill field” task into two sub tasks, “field activation” and “field value specification”.

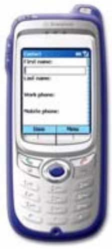


Figure 2. Contact creation form.

The current version of the contact manager application that we developed does not have any relation to the phone contact manager. It does not save the contacts in the phone contact list and it does not allow the user to find an existing contact. Our application can be generalized to a form filling task. It is composed of a main window that contains the form, as shown in Figure 2. The implemented form contains a subset of fields of the existing unimodal phone contact manager on SPV c500.

Indeed we have implemented six of the almost 50 fields that exist in the SPV c500 contact manager. Values of some of those fields are limited because of the need for vocal recognition. The following fields have been implemented:

- First Name: 10 different names
- Last name: 10 different names
- Work phone: 4 numbers
- Mobile phone: 4 numbers
- Group: {Friend, Work, Family, Other}
- City: {London, Madrid}

In Figure 3, we use the Hierarchical Task Analysis (HTA) notation to define the tasks that the phone user will be able to undertake. Nodes in the task tree are elementary tasks independent from interaction modalities. Those elementary abstract tasks will then be mapped into elementary concrete tasks using ICARE.

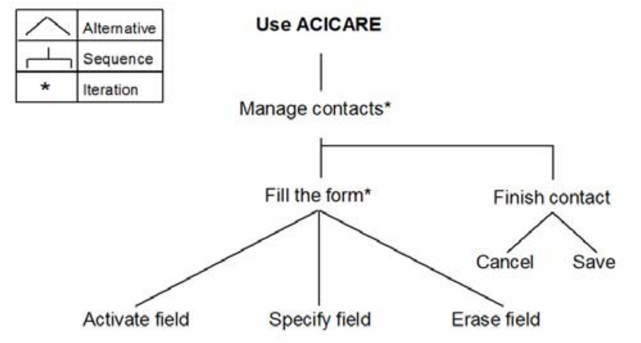


Figure 3. Hierarchical task analysis of the contact manager application.

We define the elementary tasks as follows:

- <Activate field>: Put the focus on a specified field.
- <Specify field >: Specify the value that can be numeric or textual.
- <Erase field>: Erase the content of the active field.
- <Cancel>: Close the application without creating a contact.
- <Save>: Create a new contact and close the application.

In this paper we will illustrate our ACICARE platform by considering those tasks of the contact manager application. Each of those abstract tasks is translated using ACICARE into a concrete task. We now present the ACICARE platform before illustrating the approach using this contact manager application.

4. ACICARE PLATFORM

ACICARE comes from the connection of two tools, namely ICARE and ACIDU. We will first present ICARE Mobile, a mobile implementation of the ICARE component-based platform. Then we will describe ACIDU, a probe that gathers customer's real usage on mobile phones. Finally, we will show how those tools are connected to form the ACICARE platform.

4.1 ICARE-Mobile

ICARE stands for Interaction-CARE (Complementarity, Assignment, Redundancy, Equivalence, [11]). The ICARE platform relies on a Component-Based Development (CBD) approach that offers the established advantages of reducing the production costs, and of verifying the software engineering properties of reusability maintainability and evolution [3].

ICARE [3] was originally designed to be used on PCs and was implemented using JavaBeans components. In this article, we present an ICARE implementation in C# that runs on mobile phones (Windows Mobile). The only difference from the original version is that ICARE includes an editor that enables the system designer to graphically manipulate and assemble JavaBeans components in order to create a multimodal interaction for a given task. From this graphical specification, the code for the multimodal interaction is automatically generated. As the ICARE editor is based on JavaBeans properties, it cannot be directly reused for components written in C#. For now on, components running on mobile phones are manually assembled. In the rest of the paper, we use the term 'ICARE' to mean the term 'ICARE-Mobile'.

ICARE COMPONENTS

The ICARE components are fully described in [3]. We identify two kinds of ICARE components: (1) elementary components that define "pure interaction modality" as defined in the theory of modalities [2], and (2) generic composition components that specify the combined usage of modalities.

Elementary components

Elementary components are dedicated to interaction modalities. In [11] we define an *interaction modality* as the coupling of a physical interaction device d with an interaction language L: <d, L>. A *physical device* is an artefact of the system that acquires (input device) information. Examples of such devices on mobile phones include the keyboard and microphone. A physical device as part of an interaction modality is manipulated by the phone user while interacting and does not have to be confused with the term "mobile device" that denotes the underlying whole platform (e.g., a phone or a PDA). An *interaction language* defines a set of well-formed expressions (i.e., a conventional assembly of symbols) that convey meaning. Examples of interaction languages include pseudo-natural language, direct manipulation and localization. An *interaction modality* such as speech input is then described as the couple <microphone, pseudo natural language NL>, where NL is defined by a specific grammar.

Based on this definition of an interaction modality, two types of elementary ICARE components are defined, namely Device and Interaction Language components [2]. ICARE Device components implemented on mobile phones are Keyboard Device, Dedicated Key Device (corresponding to the erase key, the navigation keys, the scroll keys, etc.) and Microphone Device. ICARE Interaction Language components implemented on mobile phones depend on the application: for we developed a "Field Content" component and a "Field Content Char by Char" component for our contact manager application.

Device and Interaction Language components constitute the building blocks for defining modalities. The designer can then combine these components in order to specify a new composed modality, in other words, a combined usage of several modalities.

Generic composition components

The CARE properties [11] characterize the different usages of multiple modalities. Based on the CARE properties, we developed four composition components running on mobile phones: Complementarity, Redundancy, Equivalence and Redundancy/Equivalence. Assignment is not explicit in an ICARE specification and is represented by a single link between two components. Indeed a component A linked to a single component B implies that A is assigned to B. As shown in Figure 4, we present an example of an ICARE specification that includes the Equivalence composition component: for specifying a field of the form, two modalities are used in an equivalent way: the phone user can fill the field by saying "Smith" or by typing "Smith" on the keyboard.

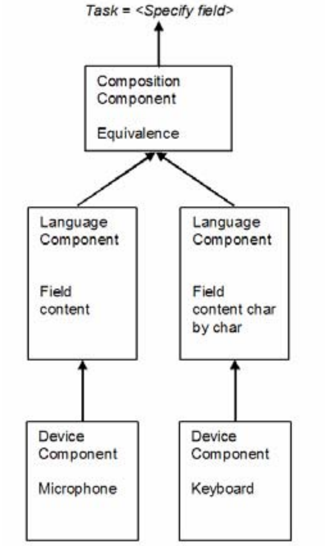


Figure 4. ICARE-Mobile specification for the task <specify field>: an example of equivalence of two modalities.

4.2 ACIDU

As explained in Section 2, in order to study the use and the usability of multimodal interaction on mobile phones, the capture of usage data in realistic situations is a necessary step. For capturing the usage data, ACICARE is based on ACIDU [4]: it is a tool implemented on mobile phones that gathers the used functionalities (e.g. camera and calendar), durations of use as well as navigation (e.g. the opened windows). In order to collect objective data (no private information, such as conversations or message contents are collected), ACIDU has been implemented as an embedded application. ACIDU is a useful tool for:

- identifying and understanding customers' real usage,

- drawing up profiles,

- supplementing the DATA usage information gathered by Orange (e.g. number of SMS sent) by collecting local usage functionalities (e.g. camera and calendar),

- capturing usage data for being analyzed as part of an iterative design approach.

In order to evaluate the technical feasibility of ACIDU, a field study has been conducted [4]. For the installation, the customers received a SMS enclosing the URL for downloading the application. They could de-install the application easily and at any time. During the field study, the log files were sent automatically by GSM or GPRS connection.

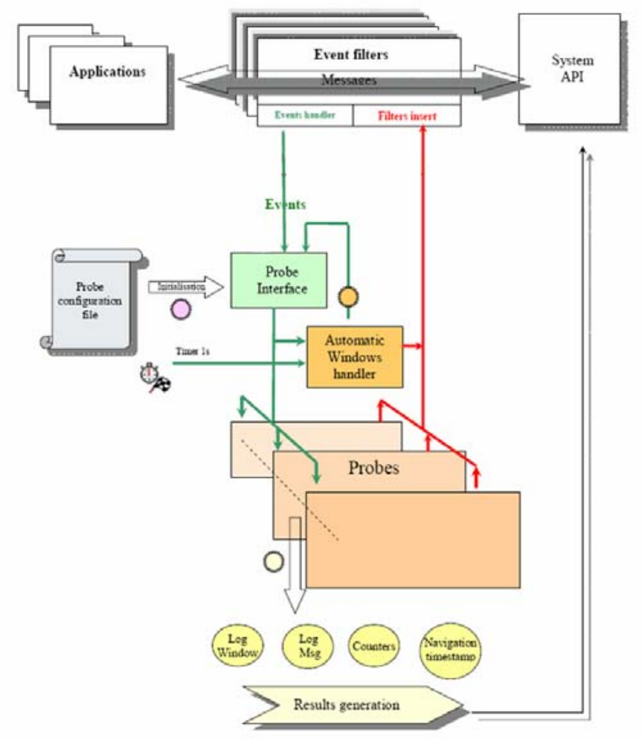


Figure 5. Overall architecture of the ACIDU version developed on Windows CE (from [3]).

The tool is implemented on Windows CE and on Symbian OS. As shown in Figure 5, all probes are derived from one object implementing "automatic windows handler" function. It means that each probe receives an event each time a new window appears on the screen. This event allows probes to focus or not on a particular application. The probe configuration file declares the events of interest for each probe. All the captured events are time stamped and gathered into a circular buffer, waiting for later upload to the server.

For multimodal interaction, the different levels of abstraction of the captured events are based on the ones of ICARE and we describe them in the next section that focuses on the connection between the two tools, ICARE and ACIDU.

4.3 Connection between ICARE and ACIDU

Each ICARE component is connected to ACIDU. When an event is received by an ICARE component, the event content and timestamp along with the name of the component are saved in a log file and ACIDU is informed of the modification of the log file by an event sent by the ICARE component as shown in Figure 6. The multimodal usage capture is therefore based on the ICARE conceptual model defining four levels of capture: device, interaction language, composition and task. Data captured by ACICARE explicitly belong to one of those levels. Data captured at each level will have the following format:

- Device level:

  (name_device, {value}, time)

- Interaction Language level:

  (**IL** name_language, input value, output value, time)

- Composition level:

(name_composition **Composition**, input value 1, input value 2… input value n, output value, time)

- Task level:

  (task_name **Task**, {task parameter}, input value, time)

Moreover, except for the lowest level (Device), the information captured at one level includes information from the lower levels, as in the Context Toolkit of A. Dey [5]. For example, if the level of capture is set to Interaction Language, we may obtain: (IL Voice Command, (Microphone, "last name", t), Field=last name, t'). In this example, the coupling of an Interaction Language component Voice Command and a Device component Microphone constitutes a modality. Although the level of capture is set to Interaction Language, information about the Device level is collected: (Microphone, "last name", t). The timestamp (t or t') corresponds to the instant the ICARE component writes the information in the log file.

When the phone user quits the application, a message is sent to ACIDU. ACIDU will send the log file to the ACIDU server. For sending the log file to the server, two strategies are possible:

- Accumulative: When the memory is full, ACIDU sends the file to the server.
- Instant: After receiving a message from ICARE, ACIDU sends data contained in the log file.

The strategy selection will depend on usage context and on technical limitations. After sending the log file, ACIDU will then manage the file deletion on the mobile phone once it has received the server reception confirmation.

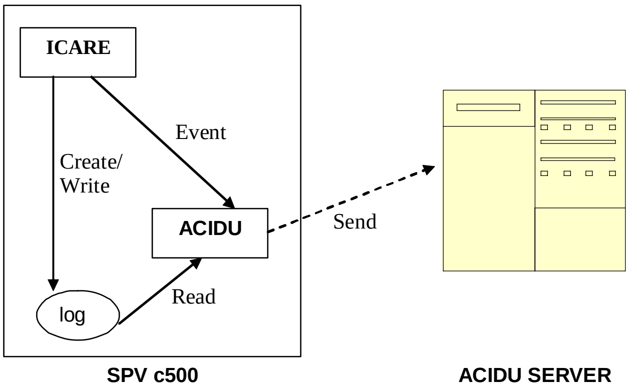


Figure 6. Connection between ICARE and ACIDU.

5. ILLUSTRATIVE EXAMPLE

In this section, we use our contact manager application developed on the SPV c500 phone for illustrating ACICARE. We first present one assembly of implemented ACICARE components that defines one design solution of multimodal interaction and then the corresponding captured usage data for a given scenario.

5.1 Multimodal Interaction and ACICARE Components

Three ICARE Device components have been implemented on the mobile phone:

- Microphone Device component
- Keyboard Device component
- Dedicated Key Device component

For implementing the speech recognition (Microphone Device component), we use an embedded solution made by Fonix Speech Inc., recognizing isolated words (speaker independent). Speech can be used for all the elementary tasks described in Section 3 (Figure 3). Figure 7 and Figure 8 show the use of speech modality for the <Activate field> and <Specify field > tasks.

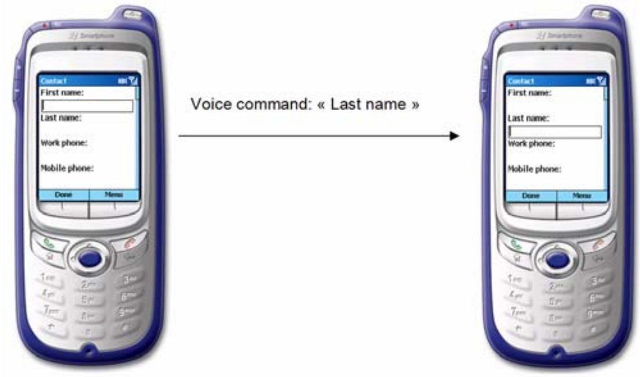


Figure 7. An example of a recognized spoken utterance for the task <Activate field>.

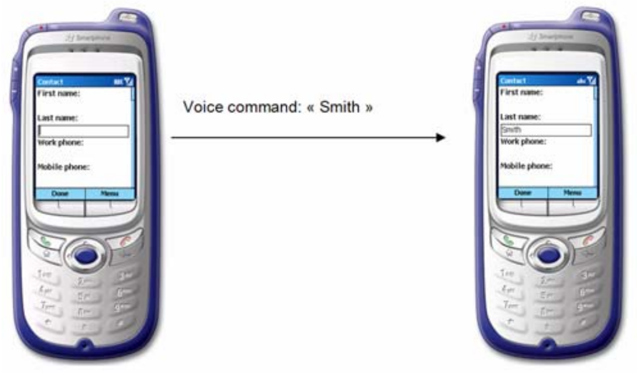


Figure 8. An example of a recognized spoken utterance for the task <Specify field>.

Since speech can be used to fill the form's fields, the size of recognized vocabulary should be quite large. For this example, it is composed of a list of ten first and last names, eight numbers composed of two digits ("twenty one" for example) and by the words of Table 1. The vocabulary can be easily changed for each test.

Table 1. Vocabulary recognized by the speech recognizer.

Name	First	Second	Phone	Mobile
Work	Group	City	Friend	Family
Other	Boston	Madrid	Next	Previous
Up	Down	Save	Cancel	Erase

Modalities used for the task "Fill form"

All the modalities for the task "Fill form" are presented in Figures 9 and 10. The abstract task "Fill form" is divided into three elementary abstract tasks: <Activate field>, <Specify field> and <Erase field> (Figure 3).

- For the <Activate field> task, we have one speech command and two keyboard commands. While speech allows a direct access to a particular field, navigation key supports navigation field by field. Moreover it is possible to initiate an automatic circular shift of field focus by a key press (a side key as shown in Figure 10) and to stop it by another key press (the stop key as shown in Figure 9).
- For the <Specify field> task, apart from the speech commands, we consider the keyboard as in the present version of the contact manager.
- For the <Erase field> task, the speech command "Erase" erases all the content of the field, while using the erase key the user can erase only one letter.

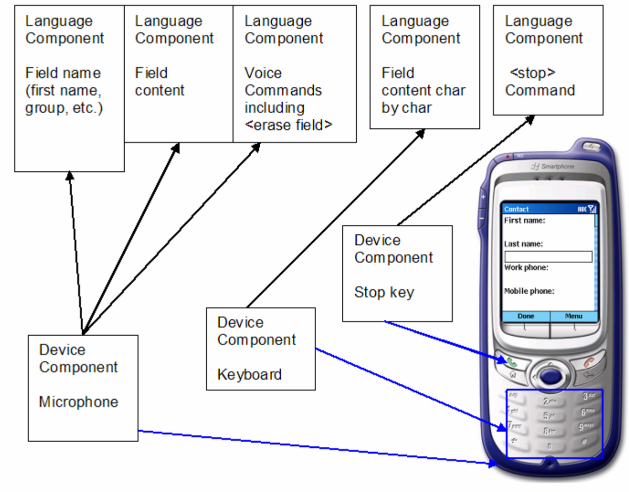


Figure 9. Input Modalities for the task "Fill form".

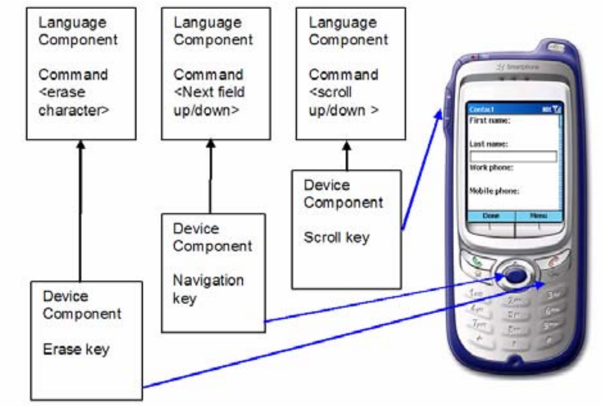


Figure 10. Input Modalities for the task "Fill form".

Modalities used for the task "Finish contact"

The abstract task "Finish contact" is divided into two elementary abstract tasks: <Cancel> and <Save> (figure 3). For these elementary tasks two modalities, based on speech and soft keys, have been implemented as shown in Figure 11.

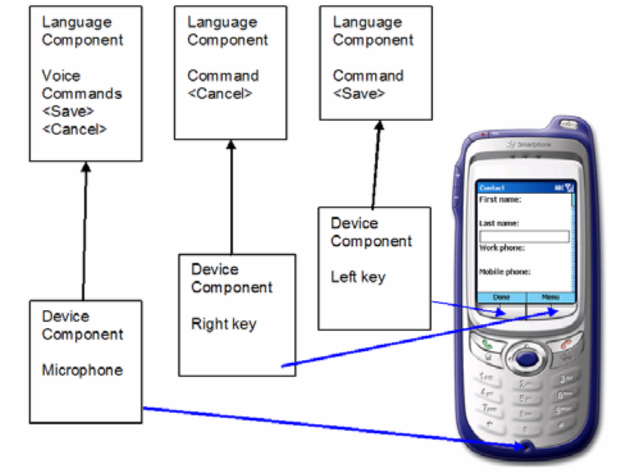


Figure 11. Input Modalities for the elementary tasks <Cancel> and <Save>.

After implementing a set of modalities as ACICARE components (Figures 9, 10 and 11), the corresponding components are composed and linked to tasks in order to design the multimodal interaction.

Figure 12 presents the ACICARE diagrams for the two elementary tasks <Specify field> and <Erase field>. In both diagrams, the speech modality and the keyboard modality are equivalent.

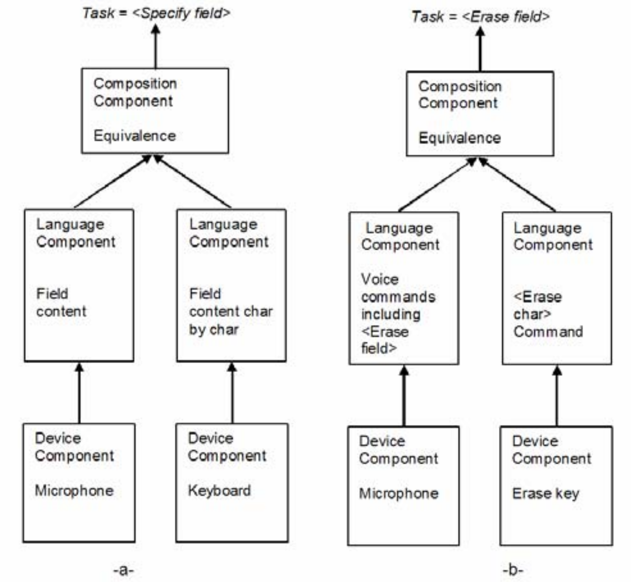


Figure 12 ACICARE diagrams for the elementary tasks <Specify field> (a) and <Erase field> (b).

Figure 13 presents the ACICARE diagram for the <Activate field> task. Three modalities are equivalent for this task: two of them are pure while the other one is composed and involves a Complementarity composition component.

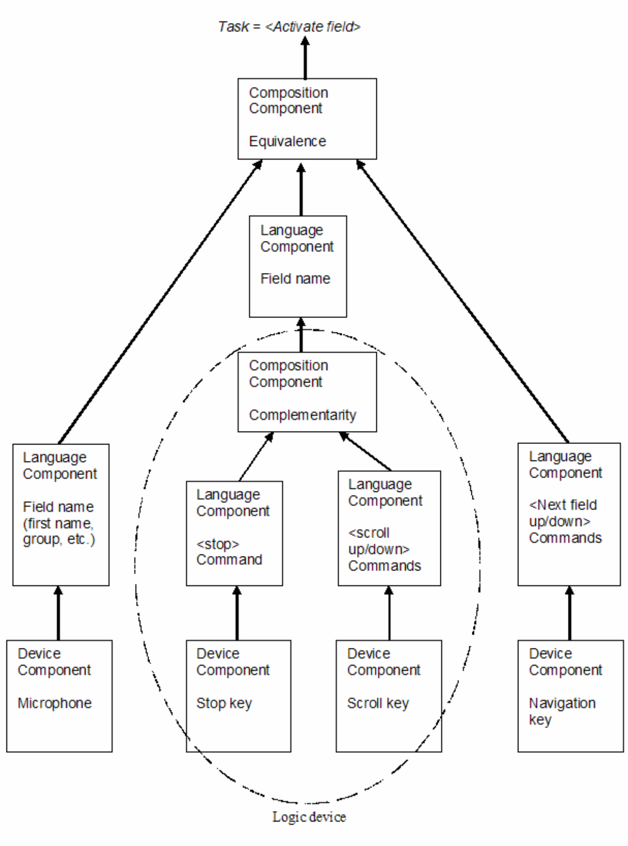


Figure 13. ACICARE diagram for the elementary task <Activate field>.

The Complementarity composition component describes the sequential but combined use of two modalities based on actions on keys: side keys for activating the automatic scroll of the fields and the stop key for selecting one field. For defining a sequential usage of the modalities, the temporal window of the Complementary composition component is set to infinite. Without receiving the two events along the two modalities, the Complementary component does not provide an output. Having the two pieces of information, the component combines them and provides output data to its connected component: <field name, stop>. Such output data needs to be translated by an Interaction Language component into a form that the <Activate field> task can understand, that is <field name>. Consequently, the composition of those two modalities can be seen as a logic device (Figure 13) that is then linked to an Interaction Language component to form a combined modality.

Finally, Figure 14 presents the ACICARE diagrams for elementary tasks <Cancel> and <Save>. As in the previous figure, the microphone modality and the keyboard modality are equivalent.

5.2 Usage Capture

Based on the design solution described by the ACICARE diagrams of Figures 12, 13 and 14, we now present an example of data capture. The format of the log file is described in Section 4.3. We consider a scenario that involves all the abstract tasks and illustrates the use of nearly all the pure and combined modalities. We suppose that when the scenario starts, the form is empty and the focus is set on the upper field, that is the "First name" field.

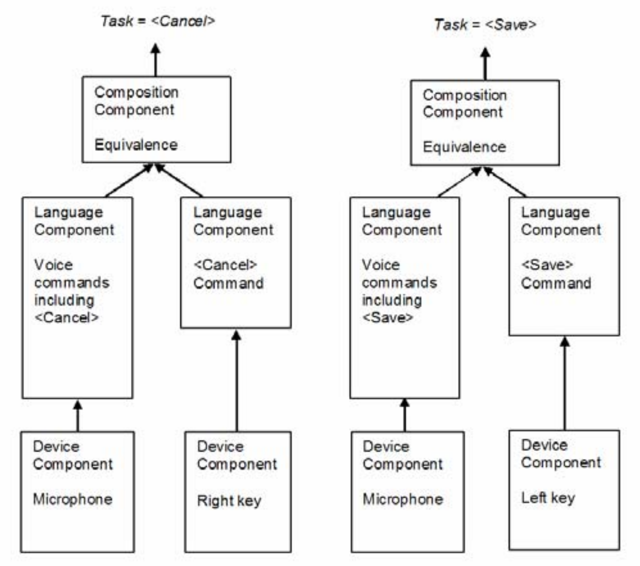
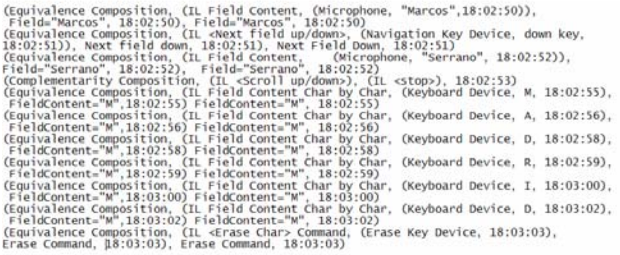


Figure 14. ACICARE diagrams for the elementary tasks <Cancel> and <Save>.

Usage scenario: A user wants to fill out the "First name" field. He says "John". He wants to fill out the "Last name" field. S/he pushes the down key and then s/he says "Smith". Finally s/he wants to specify the city field. S/he pushes the scroll down key and when the focus has reached the "City" field, s/he says "Stop". Then s/he types "Madriz" on the keyboard. S/he wants to erase the "z" and so s/he pushes the erase key. Then s/he types "d" to correctly complete the word "Madrid" in the "City" field. Finally, s/he saves the form by pushing the left key.

We present the resulting scenario log file in Figure 15: the capture level is set to Composition level. As explained in Section 4.3, we also capture data at the lower level than the Composition one (i.e., Interaction Language and Device levels).



```
(Equivalence Composition, (IL Field Content, (Microphone, "Marcos",18:02:50)),
 Field="Marcos", 18:02:50), Field="Marcos", 18:02:50)
(Equivalence Composition, (IL <Next field up/down>, (Navigation Key Device, down key,
18:02:51)), Next field down, 18:02:51), Next Field Down, 18:02:51)
(Equivalence Composition, (IL Field Content,    (Microphone, "Serrano", 18:02:52)),
Field="Serrano", 18:02:52),  Field="Serrano", 18:02:52)
(Complementarity Composition, (IL <Scroll up/down>), (IL <stop>), 18:02:53)
(Equivalence Composition, (IL Field Content Char by Char, (Keyboard Device, M, 18:02:55),
 FieldContent="M",18:02:55) FieldContent="M", 18:02:55)
(Equivalence Composition, (IL Field Content Char by Char, (Keyboard Device, A, 18:02:56),
 FieldContent="M",18:02:56) FieldContent="M", 18:02:56)
(Equivalence Composition, (IL Field Content Char by Char, (Keyboard Device, D, 18:02:58),
 FieldContent="M",18:02:58) FieldContent="M", 18:02:58)
(Equivalence Composition, (IL Field Content Char by Char, (Keyboard Device, R, 18:02:59),
 FieldContent="M",18:02:59) FieldContent="M", 18:02:59)
(Equivalence Composition, (IL Field Content Char by Char, (Keyboard Device, I, 18:03:00),
 FieldContent="M",18:03:00) FieldContent="M", 18:03:00)
(Equivalence Composition, (IL Field Content Char by Char, (Keyboard Device, D, 18:03:02),
 FieldContent="M",18:03:02) FieldContent="M", 18:03:02)
(Equivalence Composition, (IL <Erase Char> Command, (Erase Key Device, 18:03:03),
Erase Command, 18:03:03), Erase Command, 18:03:03)
```

Figure 15. Usage scenario log file at the Composition capture level.

From the analysis of those log files, information about usage includes:

- How many times a user created a new contact.
- How many times a user used Complementarity for a given task.
- The composition frequency for two given modalities.
- For which tasks speech is used.
- How many times a user pressed the Erase key: that could give information about keyboard usability.

As shown by the above list of examples, analysis can focus on the task, on the composition of modalities, on the usage of modalities or on the usage of a particular device. The analysis of the log files is part of our future work as we explain in the conclusion.

6. CONCLUSION AND FUTURE WORK

The paper presents a new tool, ACICARE that enables fast development of multimodal interaction and provides a support for evaluation of multimodal applications on mobile devices. ACICARE is dedicated to the overall iterative process by providing a way to quickly explore different design solutions and a support for continual user evaluation in realistic situations. Our ACICARE platform has been used for the implementation of a running example, for which we tested the usage capture.

Before enriching the ACICARE platform, we plan to evaluate the effectiveness of the platform for iterative design, by using it for the design of a multimodal application on mobile phones. Several design solutions will be evaluated based on the collected usage data. We then plan to extend the platform by defining a graphical tool for analysing the captured data: the tool will be based on the ACICARE levels of capture for manipulating (e.g. filtering) the data. Another extension is to develop the editor for enabling the system designer to graphically manipulate and assemble the ACICARE components (i.e the ACICARE diagrams from which the code will be automatically generated).

7. ACKNOWLEDGMENTS

This work has been partly supported by France Telecom R&D, under contract PACR Usage No 46135763 and by the SIMILAR network of excellence (http://www.similar.cc), the European research task force creating human-machine interfaces similar to human-human communication of the European Sixth Framework Programme (FP6-2002-IST1-507609). Thanks to Fonix Speech Inc. for allowing the use of their voice recognition SDK. Special thanks to G. Serghiou for reviewing the paper.

8. REFERENCES

[1] Bailie, L., and Schatz, R. Exploring Multimodality in the Laboratory and the Field. In *Proceedings of ICMI'05*. October 2005, Trento, Italy.

[2] Bouchet, J. and Nigay, L. ICARE: A Component-Based Approach for the Design and Development of Multimodal Interfaces. *Extended Abstracts CHI'04* (2004). pp 1325-1328.

[3] Bouchet, J., Nigay, L. and Ganille, T. ICARE Sofware Components for Rapidly Developing Multimodal Interfaces. *In Proceedings of ICMI'04*. ACM Press, 2004.

[4] Demumieux, R. and Losquin, P. Gathering customer's real usage on mobile phones. In *Proceedings of MobileHCI'05*. September 2005, Salzburg, Austria.

[5] Dey,A., Abowd, G. and Salber, D. Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In *HCI'01 Journal*. Vol. 16, Lawrence Erlbaum, 2001, pp. 96-166.

[6] Eisenstein, J., Vanderdonckt, J., and Puerta, A. Applying Model-Based Techniques to the Development of Uis for Mobile Computers. In *Proceedings of IUI'01*. ACM Press, 2001.

[7] Florins, M., and Vanderdonckt, J. Graceful Degradation of User Interfaces as a Design Method for Multiplatform Systems. *Proceedings of IUI'04*.

[8] Johnston, M., Bangalore, S., Vasireddy, G. et al. MATCH: An Architecture for Multimodal Dialogue System. In *Proceedings of the 40th Annual Meeting of the ACL*. pp. 376-383.

[9] Kjeldskov, J., Skov, M.B., Als, B.S. and Hoegh, R.T. Is it Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the Field. In *Proceedings of MobileHCI'04*. Springer-Verlag, Berlin, Heidelberg, 2004, 61-73.

[10] Myers, B., Hudson, S.E. and Pausch, R. Past, Present and Future of User Interface Software Tools. *In Transactions on CHI*. Volume 7, Issue 1. March 2000.

[11] Nigay, L. and Coutaz, J. The CARE Properties and Their Impact on Software Design. Intelligence and Multimodality in Multimedia Interfaces, 1997.

[12] Oviatt, S.L. Multimodal System Processing in Mobile Environments. In *Proceedings of the Thirteenth Annual ACM Symposium on User Interface Software Technology*. ACM Press, New York, 2000, pp. 21-30.

[13] Schatz, R., Simon, R., Anegg, H., et al. Developing Mobile Multimodal Applications. In Proceedings of HCI'05.

[14] Simon, R., Wegscheider, F. and Tolar, K. Tool-Supported Single Authoring for Device Independence and Multimodality. In *Proceedings of MobileHCI'05,* September 2005, Salzburg, Austria

[15] Thevenin, D. and Coutaz, J. Plasticity of User Interfaces: Framework and Research Agenda. In *Proceedings of INTERACT'99*. IOS Press, 1999.

[16] Voice Extensible Markup Language (VoiceXML) Version 2.0. W3C Recommendation 16 March 2004. http://www.w3.org/TR/voicexml20/

[17] XHTML + Voice Profile 1.0, W3C Note 21, December 2001. http://www.w3.org/TR/xhtml+voice/

[18] Zouinar, M., Salembier, P., Pasqualetti, L., et al. Part 1 Interaction: Chapter 4. Multimodal Interaction on Mobile Artefacts. Communicating with Smart Objects-Developing Technology for Usable Pervasive Computing Systems. Hermes Penton/Kogan Page Science, 2003.