ICARE: A Component-Based Approach for the Design and Development of Multimodal Interfaces

Jullien Bouchet and Laurence Nigay
CLIPS-IMAG, 38000 Grenoble, France
Jullien.Bouchet@imag.fr, Laurence.Nigay@imag.fr

Abstract
Multimodal interactive systems support multiple interaction techniques such as the synergistic use of speech, gesture and eye gaze tracking. The flexibility they offer results in an increased complexity that current software development tools do not address appropriately. In this paper we describe a component-based approach, called ICARE, for specifying and developing multimodal interfaces. Our approach relies on two types of components: (i) elementary components that describe pure modalities and (ii) composition components (Complementarity, Redundancy and Equivalence) that enable the designer to specify combined usage of modalities. The designer graphically assembles the ICARE components and the code of the multimodal user interface is automatically generated. Although the ICARE platform is not fully developed, we illustrate the applicability of the approach with the implementation of two multimodal systems: MEMO a GeoNote system and MID, a multimodal identification interface.

Categories & Subject Descriptors: H.5.2 **[Information Interfaces and Presentation]**: User Interfaces - *Input devices and strategies*, *Interaction styles*, *Prototyping*; D.2.2 **[Software Engineering]**: Design Tools and Techniques – *User Interfaces*

General Terms: Algorithms; Human Factors

Keywords: Multimodal Interactive Systems; Software Components

INTRODUCTION

The area of multimodal interaction has expanded rapidly and since the seminal "Put that there" demonstrator [3] that combines speech, gesture and eye tracking, significant achievements have been made in terms of both modalities and real multimodal systems.

Parallel to the development of the Graphical User Interface technology, natural language processing, computer vision, 3-D sound, and gesture recognition have made significant progress [11]. In addition recent interaction paradigms such as perceptual User Interface (UI) [16], tangible UI [6] and embodied UI [5] open a vast world of possibilities for interaction modalities including modalities based on the manipulation of physical objects such as a bottle and modalities based on the manipulation of a PDA and so on. We distinguish two types of modalities: the active and passive modalities. For inputs, active modalities are used by the user to issue a command to the computer (e.g., a voice command). Passive modalities are used to capture relevant information for enhancing the realization of the task, information that is not explicitly expressed by the user to the computer such as eye tracking in the "Put that there" demonstrator [3] or location tracking for a mobile user.

In addition to many modalities that are more and more robust, conceptual and empirical work on the usage of multiple modalities (CARE properties [10], TYCOON design space [8], etc.) are now available for guiding the design of efficient and usable multimodal interfaces.

Due to this conceptual and predictive progress and the availability of numerous modalities, real multimodal systems are now built in various domains including medical [13] and military ones. One of our application domains is military. We are working on multimodal commands in the cockpit of French military planes. For example while flying, the pilot can mark a point on the ground by issuing the voice command "mark" (active modality) and looking at a particular point (passive modality). Moreover multimodal interfaces are now playing a crucial role for mobile systems since multimodality offers the required flexibility for variable usage contexts, as shown in our empirical study of multimodality on PDA [18].

Although several real multimodal systems have been built, their development still remains a difficult task, Tools dedicated to multimodal interaction are currently few and limited in scope. Either they address a specific technical problem including the fusion mechanism [9] and mutual disambiguation [12], or they are dedicated to specific modalities. For instance, the Georgia Tech Gesture Toolkit GT2k is designed to support gesture recognition [17]. In this article, we address this problem of design and implementation of multimodal UI. We describe a component-based platform that enables the designer to specify multimodal interaction by assembling components, the corresponding code being automatically generated. The structure of the paper is as follows: first, we give an overview of the graphical platform called ICARE. We then focus on the ICARE components that are manipulated by direct manipulation within the platform. We finally present multimodal systems developed by assembling ICARE components.

ICARE PLATFORM

ICARE stands for Interaction-CARE (Complementarity Assignment Redundancy Equivalence). The ICARE platform enables the designer to graphically manipulate and

CHI 2004, April 24–29, 2004, Vienna, Austria.
ACM 1-58113-703-6/04/0004.

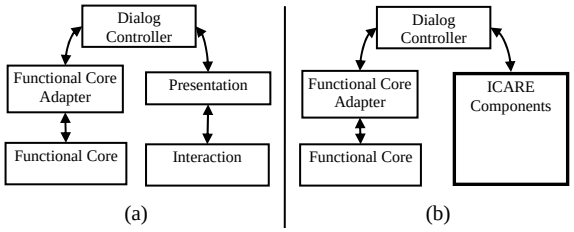

Figure 1. (a) The ARCH software architectural model. (b) ICARE components within an ARCH software architecture.

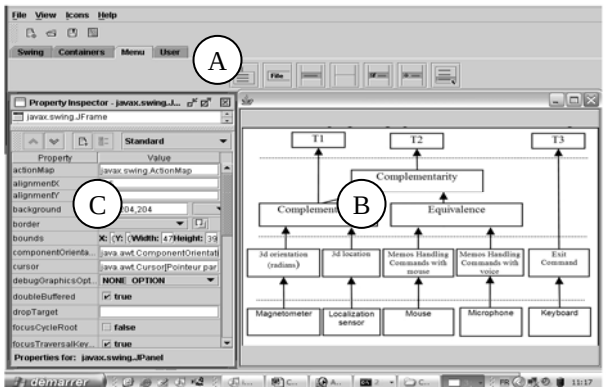

Figure 2. Sketch of the graphical ICARE platform.

assemble ICARE software components in order to specify the multimodal interaction dedicated to a given task of the interactive system under development. From this specification, the code is automatically generated. To fully understand the scope of our ICARE platform we show in Figure 1 where the automatically generated code is located within the complete code of the interactive system structured along the ARCH software architectural model [15]. Although our platform can be used for specifying inputs as well outputs, in this paper we focus on input multimodal interaction only.

The originality of the ICARE platform relies on the fact that it is dedicated to designers and not developers. Indeed the user of the ICARE platform selects the modalities and specifies the combination of modalities in terms of the CARE ergonomic properties [10], all by graphically assembling software components without knowing the details of the code of the components. From this high level specification, the code of the input multimodal UI is then generated. In addition, 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 [1].

Figure 2 presents a sketch of the user interface of the ICARE platform: it contains a palette of components (area A on Figure 2), an editing zone for assembling the selected components (area B on Figure 2) and a customization panel (area C on Figure 2) for setting the parameters of the components. Although the complete ICARE platform is not yet available, we have already designed and developed several components including modality components as well as combination components in order to validate our approach. By manually assembling these components, we have developed several multimodal systems. The following section describes these ICARE components that will in the near future be graphically manipulated in the ICARE platform.

ICARE COMPONENTS

We identify two kinds of ICARE components: (1) elementary components that enable the designer to define "pure interaction modality" as defined in the theory of modalities [2], and (2) generic composition components that enable the designer to specify combined usage of modalities. As opposed to elementary components, composition components are generic in the sense that they are not dependent on a particular modality.

Elementary components

Elementary components are dedicated to interaction modalities. In [9] we define an *interaction modality* as the coupling of a physical device d with an interaction language L: <d, L>. A *physical device* is an artifact of the system that acquires (input device) information. Examples of devices include the mouse, microphone, GPS and magnetometer. An *interaction language* defines a set of well-formed expressions (i.e., a conventional assembly of symbols) that convey meaning. The generation of a symbol, or a set of symbols, results from actions on physical devices. 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. Similarly graphic input is described in terms of <mouse, direct manipulation>. Based on this definition of an interaction modality, we identify two types of elementary ICARE components, namely Device and Interaction Language components.

An ICARE Device component represents a supplementary layer of the physical device driver. For example, the mouse Device component abstracts the data provided by the mouse driver such as button pressed/released and movement. Likewise a microphone Device component abstracts the captured signal into a recognized utterance while another microphone Device component abstracts the captured signal into a level of noise. All ICARE Device components also enrich the raw data from the device driver by adding information that include the working state of the device, the time-stamp as well as a confidence factor of the produced data, and a description of the device in terms of human manipulation (passive/active modalities, human actions involved and physical location of these actions). An ICARE Device component is then linked to a listener component, an ICARE Interaction Language component in order to form an *interaction modality*.

An ICARE Interaction Language component corresponds to the logical level of an interaction modality. For example an Interaction Language component abstracts the data from a mouse Device component into commands such as the selection of a menu option. Similarly another Interaction Language component (NL component), abstracts a set of

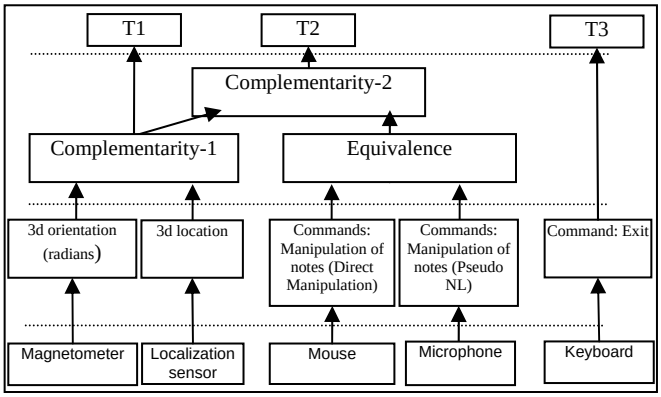

Figure 3. ICARE specification of MEMO input interaction.

characters from a microphone Device component (recognized utterance) or from a keyboard Device component into a command. A third example, shown in Figure 3, corresponds to a passive modality: the 3D Location component that abstracts data from a localization sensor (e.g., GPS) Device component into a user's location expressed in a given coordinate system. These three examples of Interaction Language components underline the fact that such components may need to rely on an external description of the well formed expressions to be obtained. Indeed in order to abstract data from the mouse into commands, a description of the graphical interface is required. Likewise NL recognition implies a description of the pseudo Natural Language to be recognized (NL grammar). Finally the 3D Location component may require a description of the environment of the user in order to produce an event such as <the user is entering a particular room>. As Device components are dependent on the underlying physical devices, Interaction Language components are dependent on a class of Device components that can produce the required inputs. For example an NL component requires a set of characters as inputs that can for example be produced by a microphone Device component or a keyboard Device component. Finally, as ICARE Device components, ICARE Interaction Language components also enrich the data by adding generic information that include the time-stamp as well as a confidence factor of the produced data.

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 [10] characterize the different usages of multiple modalities. Based on the CARE properties, we define four composition components: the Complementarity one, the Redundancy one, the Equivalence one and the Redundancy/Equivalence one. 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.

In Figure 3, we present an example of ICARE specification that includes Complementarity components. Let us consider the Complementarity-1 component. In order to compute the location and orientation of the user that is required by the application, two passive modalities are used in a complementary way. The Complementarity-1 component of Figure 3 defines a customizable temporal window for merging data received by the two Interaction Language components (respectively orientation in radians and location as latitude/longitude in WGS84 normalization and altitude in meters).

As ICARE elementary components, ICARE composition components enrich the data by adding generic information that includes the time-stamp and a confidence factor of the produced combined data. In addition composition components include parameters that the designer can fix for customizing the composition mechanism. One of these parameters is the integration strategy (as defined in our fusion mechanism [9]). For example selecting a lazy strategy for a complementary component will guarantee that only data, provided by elementary components in the same temporal window and with the highest confidence factor, are merged and sent to the next linked ICARE component

ICARE Components Implementation

For programming ICARE components, we use JavaBeans component technologies [7]. Communication between components is done by generation of events and calls of methods. Generated events transport the data to be treated. Properties of ICARE components are class attributes which can be accessed/modified (get/set). To assemble two ICARE components, it is necessary that one component subscribes to events generated by the other component.

We have developed the four ICARE composition components as well as several modality components (ICARE Device and Interaction Language components). We used the developed ICARE components for developing two multimodal systems. We manually assembled the ICARE components since the graphical ICARE platform of Figure 2 is still under development.

SYSTEMS DEVELOPED WITH ICARE COMPONENTS

A first multimodal system developed with ICARE components is MEMO, a GeoNote system [14]. MEMO allows users to annotate physical locations with digital notes which have a physical location and are then read/removed by other mobile users. To do so, MEMO supports five input active and passive modalities. Figure 3 shows the MEMO ICARE specification for input multimodal interaction. Three tasks are possible using the modalities. They define what the rest of the system receives from the ICARE components: (1) orientation and localization of the user (T1) so that the system is able to display in the HMD the visible notes according to the current position and orientation of the mobile user (2) manipulation of a note (create, pick and remove a note) (T2) and (3) exit the system (T3).

One modality "orientation" is represented by the couple magnetometer (Device component) and the three orientation angles in radians (Language component), another modality "localization" by the couple (Localization sensor, 3D location). The modalities "orientation" and "localization"

are complementary (Complementarity-1 component). Two equivalent modalities are dedicated to the manipulation of the notes: commands specified using a mouse and speech commands. One modality (Keyboard, Command) is assigned to the exit task (T3). This is a first version of MEMO: modalities and their combined usages can easily be changed. For example we are currently developing a new version with a PDA instead of a laptop. Only the Device components are changed.

Based on the ICARE specification of Figure 3, we explain how a complete command of removing the note that the user is looking at, is obtained (T2). Every three milliseconds, the modality "orientation" provides a vector of three floats corresponding to 3D orientations in radians (yaw, pitch, roll). With the same frequency, the modality "localization" provides a vector of three floats corresponding to the 3D position of the user (x, y, z). The Complementary-1 component realizes the fusion of these two vectors. With an eager strategy as soon as the vector of six floats is complete or with a lazy strategy when the temporal window is finished, an event is triggered and the vector is passed to the next component, namely Complementary-2. If the time-stamp of the event corresponding to the command <remove>, received from one of the two equivalent modalities, belongs to the same temporal window that the six float vector event belong to, the two events are combined. The component Complementary-2 then sends the complete command <remove, six parameters> to the Dialog Controller of the system that will determine the corresponding note to be removed based on the set of notes stored in the Functional Core (Figure 1).

The second multimodal system, MID (Multimodal IDentification), supports three equivalent modalities that enable the user to identify herself/himself: speech, a sequence of buttons pressed using the mouse and a password typed in using a keyboard. This second system shows that our ICARE approach allows reusability of components and therefore accelerates the development. To build MID, we reused the ICARE Equivalent component and three ICARE Device components (microphone, keyboard and mouse). And we developed new ICARE Interaction Languages components. In a few days, we obtained the final multimodal system.

FUTURE WORK

In the near future, we will complete the development of the ICARE platform that will enable the designer to graphically assemble the ICARE components. In addition we are currently developing new ICARE modality components for multimodal interaction in the Rafale (French military plane) cockpit. Finally within the ICARE platform, we plan to automatically check ergonomic properties while the designer is specifying the multimodal interaction. For example action continuity [4] can be automatically checked based on ICARE Device component properties.

ACKNOWLEDGMENTS

The work presented in the paper is partly funded by French DGA under contract #00.70.624.00.470.75.96.

REFERENCES

1. Bass, L. et al. Market Assessment of Component-Based Software Engineering. *SEI TR* (2000).
2. Bernsen, N. Modality Theory in support of multimodal interface design. *Proc. of Intelligent Multi-Media Multi-Modal Systems* (1994), 37-44.
3. Bolt, R. Put that there: Voice and gesture at the graphics interface. *Computer Graphics* (1980), 262-270.
4. Dubois, E., Nigay, L., Troccaz, J. Assessing Continuity and Compatibility in Augmented Reality Systems. *UAIS Journal*, 4 (2002), 263-273.
5. Harrison, B. et al., R. Squeeze me, Hold me, Tilt Me! An exploration of Manipulative User Interface. *Proc. of CHI'98* (1998), 17-24.
6. Ishii, H., Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *Proc. of CHI'97* (1997), 234-241.
7. JavaBeans1.01 specification, Sun Microsystems (1997), java.sun.com/products/javabeans/docs/
8. Martin, J. C. TYCOON: Theoretical Framework and Software Tools for Multimodal Interfaces. *Intelligence and Multimodality in Multimedia Interfaces*, AAAI Press (1997).
9. Nigay, L., Coutaz, J. A Generic Platform for Addressing the Multimodal Challenge. *Proc. of CHI'95* (1995), 98-105.
10. Nigay, L., Coutaz, J. Multifeature Systems: The CARE Properties and Their Impact on Software Design. *Intelligence and Multimodality in Multimedia Interfaces*, AAAI Press (1997).
11. Oviatt, S., Cohen, P. Multimodal interfaces that process what comes naturally. *Comm. of the ACM*, 43, 3 (2000), 45-53.
12. Oviatt, S. Taming recognition errors with a multimodal interface. *Comm. of the ACM*, 43, 9 (2000), 45-51.
13. Oviatt, S. et al. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. *HCI*, 15, 4 (2000), 263-322.
14. Persson, P., Espinoza, F., Cacciatore, E. GeoNote: Social Enhancement of Physical Space. *Proc. of CHI2001 Ext. Abstracts* (2001), 43-45.
15. The UIMS Tool Developers Workshop, A Metamodel for the Runtime Architecture of an Interactive System. *SIGCHI Bulletin* (1992), 32-37.
16. Turk, M., Robertson, G. Eds, Perceptual user Interfaces. *Comm. of the ACM*, 43, 3 (2000), 32-70.
17. Westeyn, T. et al. Georgia tech gesture toolkit: supporting experiments in gesture recognition. *Proc. of ICMI03* (2003), 85–92.
18. Zouinar, M. et al. Multimodal Interaction on Mobile Artifacts. *Communicating with smart objects-developing technology for usable pervasive computing systems*, Kogan Page Science (2003).