The ICARE Component-Based Approach for Multimodal Input Interaction: Application to real-time military aircraft cockpits

Jullien Bouchet, Laurence Nigay

CLIPS-IMAG lab., University of Grenoble 1
BP 53, 38041, Grenoble cedex 9, FRANCE
{jullien.bouchet, laurence.nigay}@imag.fr

Thierry Ganille

THALES-Avionics
33187 Le Haillan, FRANCE
thierry.ganille@fr.thalesgroup.com

Abstract

The area of multimodal interaction has expanded rapidly and since the seminal "Put that there" demonstrator of R. Bolt that combines speech, gesture and eye tracking, significant achievements have been made in terms of both modalities and real multimodal systems. Nevertheless the design and development of multimodal systems still remains a difficult task. In this article, we present a graphical platform called ICARE (Interaction CARE -Complementarity Assignment, Redundancy and Equivalence-) for designing and developing multimodal input interaction. As part of the INTUITION project, we will explain how we design and develop multimodal input interaction using our tool ICARE for military aircraft cockpit in a real-time simulator of a military fighter.

1 Introduction

Multimodal user interfaces support multiple interaction modalities, which may be used sequentially or concurrently, and independently or combined synergistically (Nigay & Coutaz, 1993). In this article we focus on input multimodal interaction: multiple modalities and forms of multimodality available to the user for specifying commands to the interactive system. Since the seminal "Put that there" demonstrator (Bolt, 1980) that combines speech, gesture and eye tracking, significant achievements have been made in terms of both modalities and real multimodal systems. Indeed, in addition to more and more robust modalities, conceptual and empirical work on the usage of multiple modalities is now available for guiding the design of efficient and usable multimodal interfaces. As a result, real multimodal systems are now being built in various application domains including medicine (Oviatt et al., 2000) and education. As part of the INTUITION project, our application domain for input multimodal interaction is military and we focus on input multimodal interaction for military aircraft cockpits.

Although several real multimodal systems have been built, their development still remains a difficult task. The power and versatility of multimodal interfaces result in an increased complexity that current design methods and tools do not address appropriately. Tools dedicated to multimodal interaction are currently few and limited in scope. Either they address a specific technical problem including the fusion mechanism (Nigay & Coutaz, 1995), the composition of several devices (Dragicevic, P. & Fekete, J.-D., 2002) and mutual disambiguation (Oviatt, 2000), or they are dedicated to specific modalities such as gesture or speech. In this article we address this problem of design and implementation of multimodal input user interfaces. We describe a component-based platform, ICARE, that enables the designer to specify multimodal interaction by assembling components, the corresponding code being automatically generated.

The structure of the article is as follows: first we present our ICARE platform by explaining the underlying conceptual model and the graphical tool for designing multimodal input interaction. We then illustrate the usage of our ICARE development tool by explaining how we used it for the design and development of multimodal input interaction in INTUITION Avionic Simulator (IAS), a real-time flight simulator of a military fighter.

2 ICARE Platform

ICARE stands for Interaction-CARE (Complementarity Assignment Redundancy Equivalence). The ICARE platform enables the designer to graphically manipulate and 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. In order to present our ICARE platform, we first explain the underlying ICARE conceptual model by presenting the types of components manipulated by the platform. We then present the graphical editor that enables the design and the code generation of multimodal input interaction.

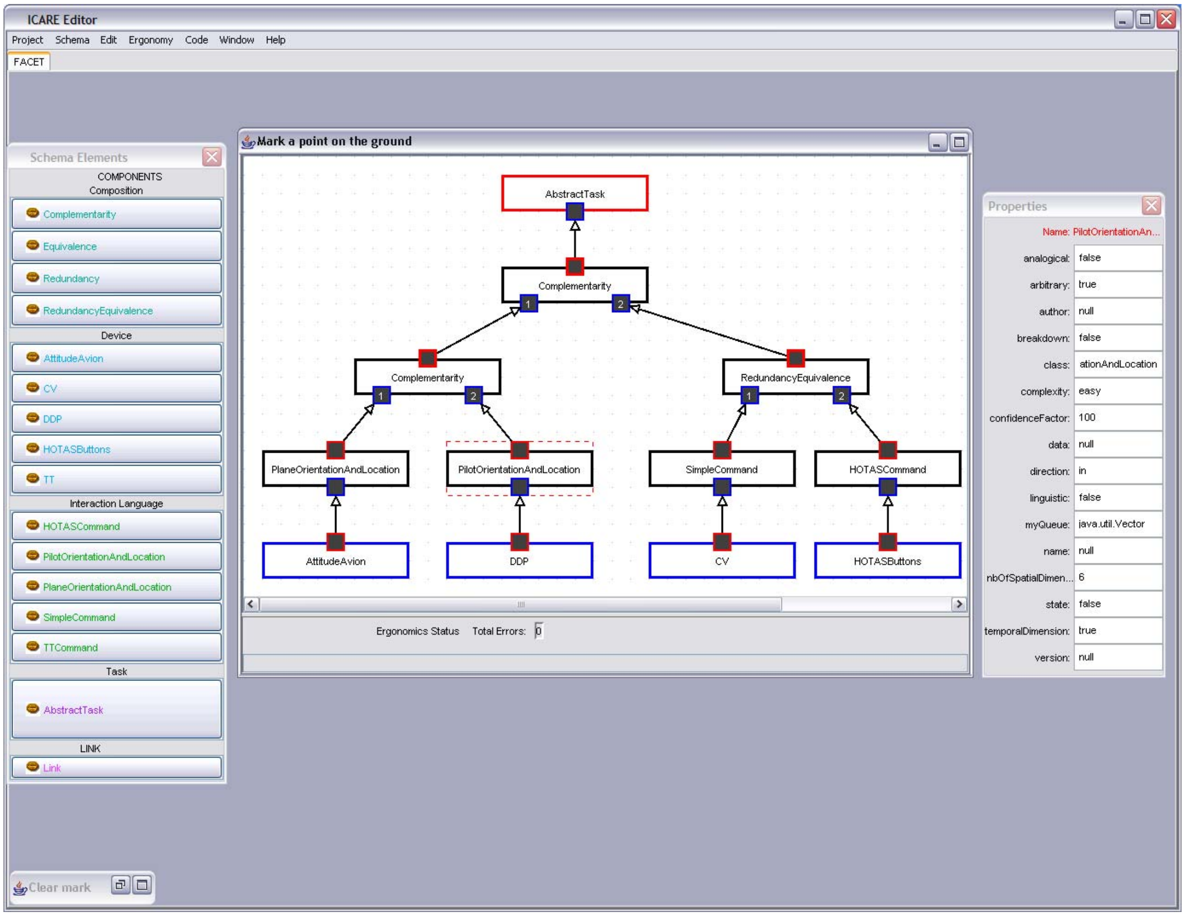

Figure 1: ICARE graphical platform

2.1 ICARE conceptual model

The ICARE conceptual model includes elementary and composition components. They are fully described in (Bouchet & Nigay, 2004), while their developments are described in (Nigay, Bouchet & Ganille, 2004).

Elementary components are building blocks useful for defining a modality. Two types of ICARE elementary components are defined: Device components and Interaction Language components. We reuse our definition of a modality (Nigay & Coutaz, 1993) as the coupling of a physical device d with an interaction language L: <d, L>. In (Nigay & Coutaz, 1997), we demonstrate the adequacy of the notions of physical device and interaction language for classifying and deriving usability properties for multimodal interaction, while in (Nigay & Coutaz, 1995) we adopt a complementary perspective and examine the relevance of these notions for software design.

Composition components describe combined usages of modalities and therefore enable us to define new composed modalities. As opposed to ICARE elementary components, Composition components are generic in the sense that they are not dependent on a particular modality. The ICARE composition components are defined based on the four CARE properties (Nigay & Coutaz, 1997): the Complementarity, Assignment, Redundancy, and Equivalence that may occur between the modalities available in a multimodal user interface. While Equivalence and Assignment express the availability and respective absence of choice between multiple modalities for performing a given task, Complementarity and Redundancy describe relationships between devices, languages or more generally between modalities for performing a given task. Because the CARE properties have been shown to be useful concepts for the design and evaluation of multimodal interaction (Nigay & Coutaz, 1997), we decided to reuse those concepts to make them explicit during the software development. We define three Composition components in our ICARE conceptual model: the Complementarity one, the Redundancy one, and the Redundancy/Equivalence one. Assignment and Equivalence are not modeled as components in our ICARE model. Indeed, an assignment is represented by a single link between two components. An ICARE component A linked to a single component B implies that A is assigned to B. As for Assignment, Equivalence is not modeled as a component. When several components (2 to n components) are linked to the same component, they are equivalent. The ICARE Composition components will be illustrated in the context of the IAS system (section 3).

2.2 ICARE graphical editor

Figure 1 presents the user interface of the ICARE platform: it contains a palette of components, an editing zone for assembling the selected components and a customization panel for setting the parameters of the components. The palette of components is organized along the three types of ICARE components: Composition, Interaction Language and Device components. The user of the ICARE platform selects the modalities (Device and Language components) and specifies the combination of modalities by selecting a Composition component, all by graphically assembling software components without knowing the details of the code of the components. After graphically specifying the multimodal interaction for a given task, the user of the ICARE platform can generate the code of the corresponding multimodal interaction.

2.3 Multimodal systems developed using ICARE

Using our ICARE platform, we developed several multimodal systems:

- A Multimodal IDentification system (MID) combining speech, keyboard and mouse modalities for identification.
- A mobile system MEMO: MEMO allows users to annotate physical locations with digital notes which have a physical location and are then read/removed by other mobile users. Two versions are currently running, one using a PDA and another one using a Head-Mounted Display. The development of MEMO based on ICARE is described in (Bouchet & Nigay, 2004).
- A Yellow Pages system, allowing the user to specify queries using speech, keyboard and mouse modalities. In addition the user can specify synergistic commands combining speech and direct manipulation. For example the user can issue the voice command “Zoom here” while selecting a point on the displayed map using a stylus.

In this article, we focus on a larger system, IAS, a flight simulator of a military fighter.

3 Multimodal interaction for military aircraft cockpits

As part of the INTUITION project, we use our ICARE platform for designing and developing the multimodal input interaction for a fighter flight simulator. We first describe the main characteristics of IAS in terms of input modalities. After presenting the global software architecture, we then focus on three tasks that we developed using the ICARE platform: <Switch equipment on/off>, <Change mode> and <Mark a point on ground>. For the three considered tasks, we explain the ICARE specification and the extensions that can be done, highlighting the benefits of our ICARE approach.

3.1 IAS : A fighter flight simulator

The INTUITION Avionic Simulator (IAS) is a real-time development flight simulator of a military fighter. IAS is used for studying future interaction techniques that could be embodied in the cockpit. Figure 2 shows a pilot testing the IAS. Examples of tasks include:

- *Pilot the aircraft*.
  The pilot must follow a predefined trajectory while adapting it according to various parameters including meteorological/geographical conditions as well as aircraft characteristics.
- *Navigate*.
  The pilot must create or modify the flight plan for example by entering new waypoints.

- *Manage the aircraft*.
  The pilot must constantly check aircraft systems (fuel, hydraulic, electrical…).
- *Manage the armory system*.
  The pilot must be able to change armory depending on the mission.
- *Manage the mission*.
  For example the pilot must be able to protect herself/himself against attacks.
- *Communicate with air traffic controllers and with other pilots of the patrol*.

Figure 2: IAS a real-time flight simulator of a military fighter

While the above listed tasks are very general, more concrete elementary sub-tasks can be identified as part of a hierarchical task analysis process. For example, one task <mark a point on ground> can be a sub-task of the general task <manage a mission>. Having identified elementary tasks (or sub-tasks), we consider the various input modalities and forms of multimodality that can be developed using ICARE for performing such elementary tasks. To do so, a set of input modalities based on different input devices is available within IAS:

- The HOTAS (Hands On Throttle And Stick) are made of two command joysticks (one for each hand) to pilot the aircraft and to issue commands such as marking a point on the ground.
- The helmet visor, which allows the pilot to select a target in the real world that depends on the orientation and location of both the pilot and the aircraft.
- Speech inputs for issuing commands such as marking a point on the ground.
- A tactile surface in between the legs of the pilot for specifying commands by direct manipulation, such as switching equipment on.

In addition to the various input modalities available within IAS, several contexts related to the military mission are clearly identified. Multimodal interfaces, by the flexibility they offer, can play a central role for managing such contexts. Indeed the pilot can change input modalities as well as forms of multimodality according to the current context, the level of stress and the physical parameters (e.g., exposure to strong accelerations). To conclude IAS defines a very rich and promising case study for input multimodality.

3.2 ICARE code within the software architecture of the entire system

In Figure 3, we present the global software architecture of the entire interactive system in order to highlight the scope of the code generated by the ICARE platform. Such architecture is detailed in (Bastide et al., 2005). Within the architecture, we identify two types of link between the ICARE code and the rest of the system:

- As pointed out in Figure 3, the connection between the ICARE code and the rest of the interactive system is at the level of the elementary tasks. From actions performed by the pilot along various modalities, the ICARE code is responsible for defining elementary tasks that are independent of the used modalities. Such elementary tasks are then transmitted to the Dialog Controller, described by a Petri Net (Bastide et al., 2005). The communication between ICARE code and the Dialog Controller has been implemented using JavaRMI. Such architecture makes the Dialog Controller independent of the interaction modalities. As a consequence, modifying the input/output modalities of the system will not imply change to the Dialog Controller.

- The ICARE code also needs to share some information with the output user interface (Rousseau, Bellik, Vernier & Bazalgette, 2005). Indeed in order to abstract low level events specified by the pilot, ICARE components need to know the graphical user interface. For example, when the pilot is selecting a button on the tactile surface, the event (x,y) must be translated into a selection of a button. Such a description of the displayed graphical objects is shared between ICARE and the output interface.

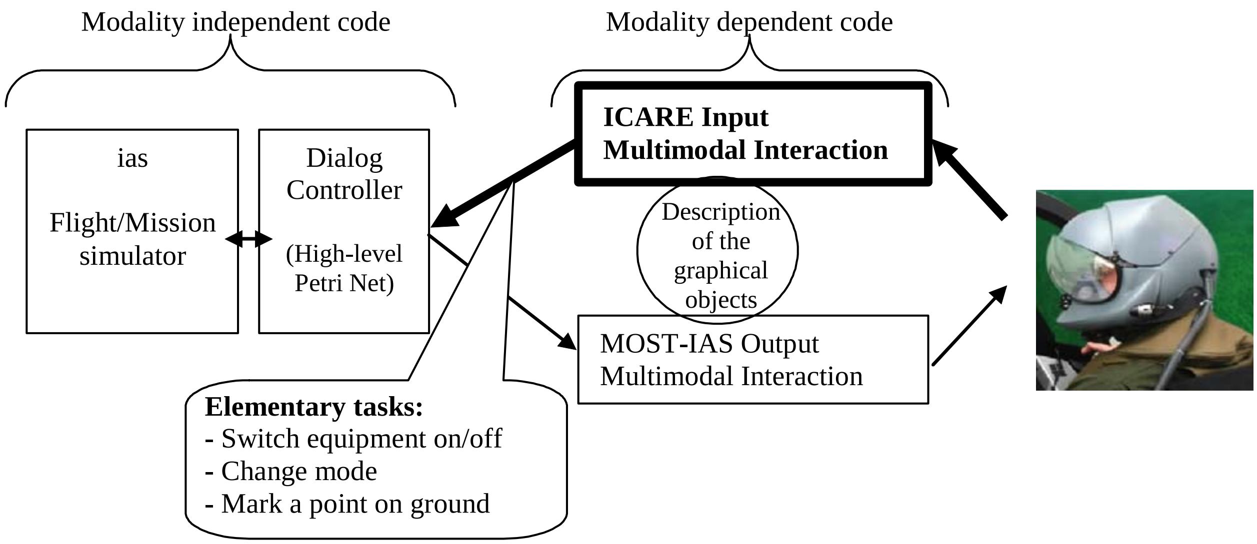

Figure 3: Multimodal Interaction for military aircraft cockpit: Global architecture

Having explained the scope of the code generated by the ICARE platform, we now focus on the input multimodal interaction for three tasks performed by the pilot.

3.3 ICARE specification and code generation for three tasks within IAS

For each task, we specify the multimodal interaction using our ICARE platform. Using the graphical editor (Figure 1), the designer defines the modalities and the relationships between the modalities by graphically assembling the ICARE components. The resulting ICARE specification describes an ICARE component chain corresponding to a pipeline from user's actions to elementary tasks understandable by the Dialog Controller. So the last ICARE component of a chain must communicate the elementary task to the Dialog Controller. Such a link has been implemented by using JavaRMI but other techniques could be used including direct call of methods, TCP/IP, UDP and so on. Having defined an ICARE component chain using the graphical editor, the code is automatically generated.

3.3.1 Task 1: Switch equipment on/off

We first consider the task of switching equipment on/off. The ICARE specification of Figure 4-a is quite simple since only one modality can be used for switching equipment on/off: the pilot selects a button displayed on the tactile surface. On the tactile surface five graphical buttons corresponding to five pieces of equipment are displayed: Radar, CME (Electronic Counter Measures), MIDS (Multifunctional Information Distribution System), SNA (Navigation and Weapon Systems) and HUD (Head Up Display). For implementing such an interaction, we used two ICARE components:

- One Device component, corresponding to the tactile surface. This ICARE Device component communicates with the physical device using TCP/IP.
- One Language component. The ICARE Language component abstracts the events from the device in terms of 10 elementary tasks <Switch the five pieces of equipment on/off > that will in turn be transmitted to the Dialog Controller. For example the Dialog Controller will receive the elementary task <Switch Radar on> without knowing how the pilot specified it. In order to be able to abstract the events into elementary tasks, the ICARE

Language component needs to have access to a description of the displayed buttons. Such a description is a resource file for the ICARE Language component.

Using the ICARE platform and having built the ICARE specification of Figure 4-a thanks to the graphical editor (Figure 1), the corresponding code is automatically generated and embedded into IAS. As an extension, we could envision letting the pilot switch equipment on/off using speech as well. Such a modification of the input interaction can easily be performed by editing the first ICARE specification (Figure 4-a) and adding two components corresponding to the speech modality, as shown in Figure 4-b.

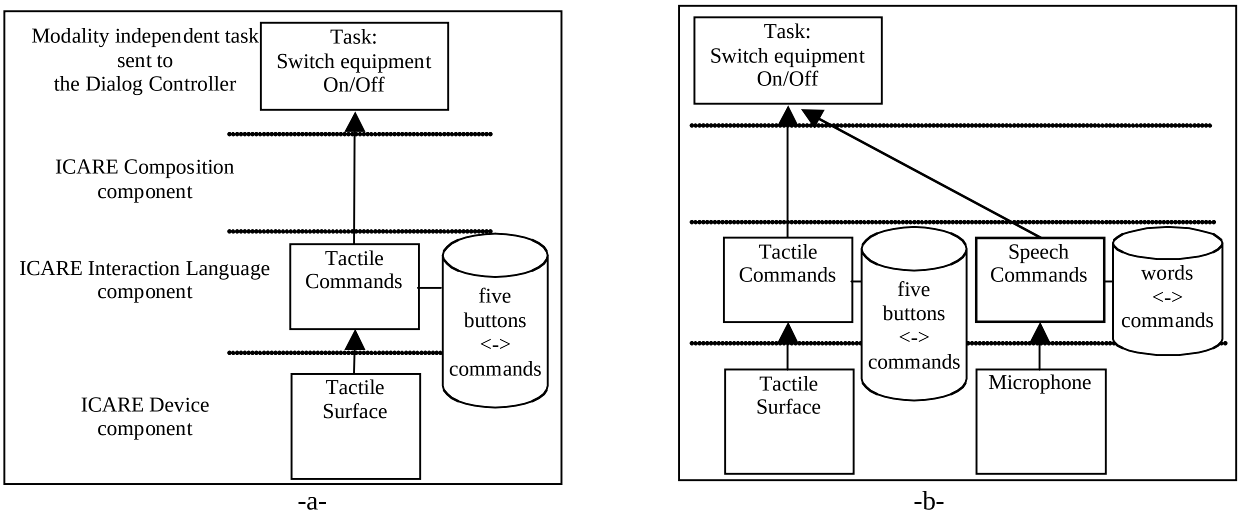

Figure 4: -a- ICARE schema for the task <Switch equipment on/off> -b- ICARE schema: Adding a new modality for performing the same task <Switch equipment on/off>

Having two equivalent modalities for performing the task (Figure 4-b), we can also envision a new extension by considering that the two modalities may be used in a redundant way. Redundancy (Nigay & Coutaz, 1997) corresponds to the case where two modalities convey redundant pieces of information that are close in time. In such a case, one of the two user's actions must be ignored. For example if the pilot selects the button Radar-on, on the tactile screen while issuing the voice command "Radar on", only one task <Switch Radar on> will be transmitted to the Dialog Controller. As it is currently specified in Figure 4-b, the Dialog Controller will receive twice the task <Switch Radar on>. To specify such potential redundant usage of the two modalities, one ICARE Composition component "Redundancy/Equivalence" must be added as shown in Figure 5. This extension is therefore very easy to do using ICARE.

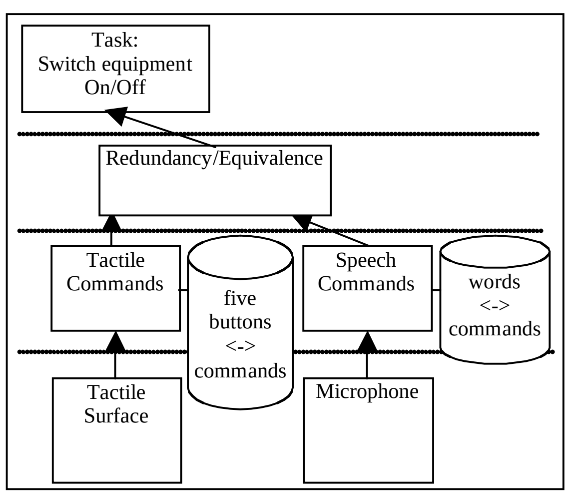

Figure 5: ICARE specification of a redundant/equivalent usage of two modalities for the task <Switch equipment on/off>

3.3.2 Task 2: Change Mode

We now consider a second elementary task <Change Mode>. The pilot can change the mode of the SNA (Navigation and Weapons Systems) as well the mode of the radar. For example the SNA can be switched to AirToAir or to AirToGround. Three modes are possible for the SNA (AirToAir, AirToGround and Navigation), while two modes are available for configuring the radar: single versus multiple targets tracking. As for the previous task <Switch equipment on/off>, only one modality has been developed and embedded in IAS for performing the <Change Mode> task: pressing HOTAS buttons. We therefore assign the HOTAS buttons to the task <Change mode> as shown by the ICARE specification of Figure 6. The HOTAS Device component is communicating with the physical device using UDP. The HOTAS Commands Language component receives the number of the button pressed by the pilot and transforms it into a mode such as (Mode = AirToAir). The new mode or elementary task is then sent to the Dialog Controller. As an extension, we could envision using two redundant modalities, voice commands and commands specified by pressing HOTAS buttons for changing the mode. Redundant usage can be justified by the criticality of the task, changing from a navigation mode to a fighting mode. This extension can be easily performed by adding a Redundancy Composition component within the ICARE specification. In such a case shown in Figure 7, the current mode will be changed only if the pilot issues the speech command while pressing the right HOTAS button.

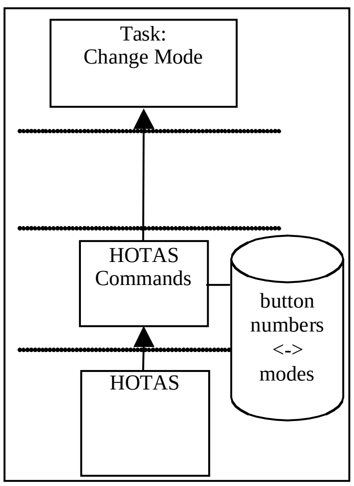

Figure 6: ICARE schema for the task <Change Mode>

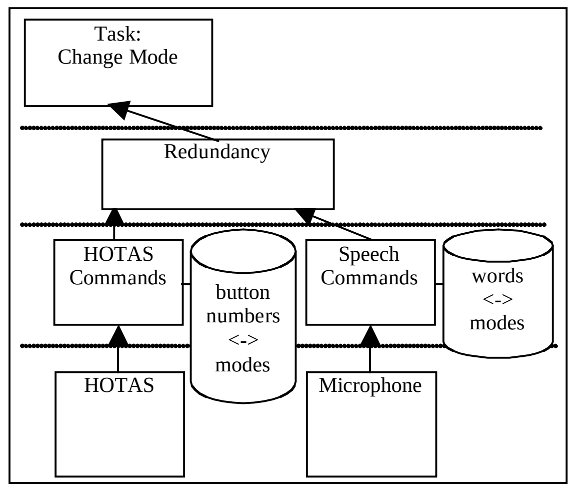

Figure 7: ICARE schema: Redundant usage of two modalities for the task <Change Mode>

3.3.3 Task 3: Mark a point on ground

The last task that we developed within IAS using the ICARE platform is the marking of a point on ground. This task is fully described in (Bouchet, Nigay & Ganille, 2004). Figure 8 shows the assembly of components for the task. Four modalities are implemented:

- M1= (Aircraft Orientation&Location tracker, Aircraft Orientation&Location)
- M2 = (Pilot's Orientation&Location tracker, Pilot's Orientation&Location)
- M3= (HOTAS, HOTAS Commands)
- M4 = (Microphone, Speech Commands)

M1 and M2 are used in a complementary way for detecting the target selected by the pilot using the helmet visor: Complementarity-1 Composition component. In addition for specifying a marking command, the pilot has the choice between two modalities, M3 and M4 that are functionally equivalent but can also be used in a redundant way thanks to the Redundancy/Equivalence Composition component. As a consequence if the pilot is pressing the HOTAS button while speaking, one point will be marked and if the pilot presses the HOTAS button and later issues the voice command (Mark), two commands will be transferred to the Dialog Controller. Finally in order to obtain a complete marking command, the command (Mark) must be combined (Complementarity-2 Composition component) with the target point defined by the pilot using the helmet visor. The obtained complete command is then sent to the Dialog Controller. The latter receives the complete task without knowing how the pilot specified it.

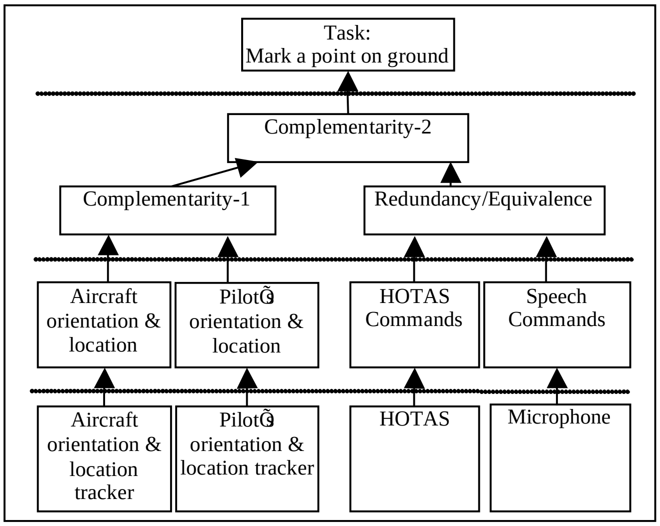

Figure 8: ICARE schema for the task <Mark a point on ground>.
Adapted from (Bouchet, Nigay & Ganille, 2004, Figure 10).

For this task, we also performed several tests for validating our ICARE approach for a real-time system. For the Redundancy/Equivalence component we fixed the temporal window (Δt) equal to 100 ms while for the Complementarity component we fixed Δt equal to 10 ms. We measured the processing time of each implemented ICARE component in IAS. We performed 10000 tests for the ICARE Modality (Device and Language) components and 300 tests for the ICARE Composition components. In particular Complementarity and Redundancy/Equivalence components were satisfactory with regards to the real-time constraints of IAS. The tests as well as the results are fully described in (Bouchet, Nigay & Ganille, 2004).

4 Summary and future work

In this article, we have presented our ICARE conceptual model that includes elementary input modality dependent components as well as generic components (reusable components) for combining modalities (fusion mechanism). We have then exposed our graphical ICARE platform that enables the designer to graphically manipulate and assemble such ICARE software components in order to specify the multimodal input interaction dedicated to a given task of the interactive system under development. From this specification (ICARE schema), the code of the multimodal interaction is automatically generated. The user of the ICARE platform selects the modalities (Device and Language components) and specifies the combination of modalities by selecting a Composition component, all by graphically assembling software components without knowing the details of the code of the components. We have illustrated the usage of the ICARE platform, by explaining how we developed the multimodal input interaction for three tasks in the IAS system, a real-time fighter flight simulator.

For the ICARE graphical platform, we identify two types of target users: the developer of interaction modalities and the designer of multimodal interaction.

- The developer is responsible for enriching the set of components managed by the platform (Figure 1) by adding new Device and Language components. The developer could also decide to provide encapsulated components, for example a complete modality component (that includes a Device component and a Language component). For adding a modality to the platform, the developer needs to encapsulate the corresponding code into an ICARE component. It is our approach in the IAS system since all the modalities have been developed by THALES specifically for military aircraft cockpits.
- The designer is defining multimodal interfaces by directly manipulating components using the mouse. It does not need to understand the details of the component code. As opposed to existing JavaBeans editors including BeanBuilder and Jbuilder, our ICARE platform makes explicit the notions of device as well as language and the CARE properties whose adequacy for studying the usability has been demonstrated (Nigay & Coutaz,

1997). Our approach therefore enables the quick development of various forms of multimodal interaction to be then experimentally tested by the end-users.

As future work, we plan to extend the graphical ICARE platform in order to automatically check ergonomic properties while the designer is specifying the multimodal interaction. For example action continuity (Dubois, Nigay & Troccaz 2002) can be automatically checked based on ICARE Device component properties and the global consistency in the usage of modalities can also be checked. As part of the INTUITION project, our next phase is to use ICARE for another application domain, namely Air Traffic Control.

5 Acknowledgments

The work presented in the article is partly funded by DGA (French Army Research Dept.) under contract INTUITION #00.70.624.00.470.75.96 and by the SIMILAR European FP6 network of excellence dedicated to multimodality (http://www.similar.cc). Special thanks to G. Serghiou for reviewing the article.

6 References

Bastide, R., Nigay, L., Bazalgette, D., Bellik. Y. & Nouvel, C. (2005). The INTUITION design process: Structuring Military Multimodal Interactive Cockpits Design According to the MVC Design Pattern. In *Conference Proceedings of HCI International 2005, the 11th International Conference on Human-Computer Interaction*, Las Vegas, Nevada, USA, July 2005. Lawrence Erlbaum Associates.

Bastide, R., Barboni, E., Bazalgette, D., Lacaze, X., Navarre, D., Palanque, P. & Schyn, A. (2005). Supporting Intuition through formal specification of the User Interface for military aircraft cockpits. . In *Conference Proceedings of HCI International 2005, the 11th International Conference on Human-Computer Interaction*, Las Vegas, Nevada, USA, July 2005. Lawrence Erlbaum Associates.

Bolt, R. (1980). Put-that-there: Voice and gesture at the graphics interface. In *Conference Proceedings of SIGGRAPH'80, the 7th annual conference on Computer graphics and interactive techniques*, Seattle, Washington, USA, July 1980 (pp. 262–270). ACM Press.

Bouchet, J., Nigay, L., & Ganille, T. (2004). ICARE Software Components for Rapidly Developing Multimodal Interfaces. In *Conference Proceedings of ICMI'2004, the Sixth International Conference on Multimodal Interfaces*, State College, Pennsylvania, USA, October 2004 (pp. 251-258). ACM Press.

Bouchet, J. & Nigay, L. (2004). ICARE: A Component-Based Approach for the Design and Development of Multimodal Interfaces. In *Conference Proceedings of ACM-CHI'04, Extended abstracts of the 2004 conference on Human factors and computing systems*, Vienna, Austria, April 2004 (pp. 1325-1328). ACM Press.

Dubois, E., Nigay, L., & Troccaz, J. (2002) Assessing Continuity and Compatibility in Augmented Reality Systems. *International Journal on Universal Access in the Information Society, Special Issue on Continuous Interaction in Future Computing Systems*, 4, 263-273. Springer Verlag.

Dragicevic, P. & Fekete, J.-D. (2002) ICON: Input Device Selection and Interaction Configuration, *Demonstration, UIST 2002 Companion*, 47-48.

Nigay, L. & Coutaz, J. (1997). The CARE Properties and Their Impact on Software Design. In *Intelligence and Multimodality in Multimedia Interfaces, Chapter 9.* AAAI Press Publ.

Nigay, L. & Coutaz, J. (1995). A Generic Platform for Addressing the Multimodal Challenge. In *Conference Proceedings of ACM-CHI'95, the SIGCHI conference on Human factors in computing systems,* Denver, Colorado, USA, May 2005 (pp. 98-105). ACM Press.

Nigay, L., & Coutaz, J. (1993). A design space for multimodal interfaces: concurrent processing and data fusion. In *Conference Proceedings of INTER-CHI'93, , the SIGCHI conference on Human factors in computing systems*, Amsterdam, Netherlands, April 1993 (pp. 172-178). ACM Press.

Oviatt, S. (2000). Taming recognition errors with a multimodal interface. *Communications of the ACM*, 43, 9, 45-51.

Oviatt, S., Cohen, P, Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. & Ferro, D. (2000). Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. Journal HCI, Human-Computer Interaction, 15, 4, 263-322. Lawrence Erlbaum Associates.

Rousseau, C., Bellik, Y., Vernier, F., & Bazalgette, D. (2005). Multimodal Output Simulation Platform for Real-Time Military Systems. In *Conference Proceedings of HCI International 2005, the 11th International Conference on Human-Computer Interaction*, Las Vegas, Nevada, USA, July 2005. Lawrence Erlbaum Associates.