The INTUITION Design Process: Structuring Military Multimodal Interactive Cockpits Design According to the MVC Design Pattern

Rémi Bastide[4], Laurence Nigay[3], Didier Bazalgette[1], Yacine Bellik[2], Christian Nouvel[5]

[1] General Delegation for the Armament
Ministry of Defense, Paris, France
didier.bazalgette@dga.defense.gouv.fr

[3] CLIPS-IMAG lab., University of Grenoble 1
BP 53, 38041, Grenoble cedex 9, France
laurence.nigay@imag.fr

[2] LIMSI-CNRS lab., University of Paris XI
BP 133, 91403, Orsay, France
yacine.bellik@limsi.fr

[4] LIIHS - IRIT lab., University of Toulouse 3
118, route de Narbonne, 31062 Toulouse, France
Remi.Bastide@irit.fr

[5] THALES-Avionics
33187 Le Haillan, France
Christian.Nouvel@fr.thalesgroup.com

Abstract

This article is concerned with the design and implementation of multimodal user interfaces. The use of multiple modalities such as vision, speech and gesture opens a vast world of possibilities in user interface design. Although the potential for innovation is high, the current understanding of how to design and build multimodal user interfaces is still primitive. The power and versatility of multimodal interfaces result in an increased complexity that current design methods and tools do not address appropriately. In this article we present the INTUITION design process for multimodal interaction. We then present the tools that are dedicated to several phases of our design process. In particular we focus on the tools for developing multimodal interfaces and the resulting software architecture of a developed multimodal interface. In the INTUITION project, our main application domain is military and we focus on the design and development of multimodal interaction for military aircraft cockpits in FACET, a real-time simulator of the French military aircraft RAFALE.

1 Introduction

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. The use of multiple modalities such as vision, speech and gesture opens a vast world of possibilities in user interface design. As a result, real multimodal interactive systems are now being built in various application domains including education and medicine. As part of the INTUITION project, our main application domain is military and we focus on the design and development of multimodal interaction for military aircraft cockpits in FACET, a real-time simulator of the French military aircraft RAFALE.

The goal of multimodal interfaces is to extend the sensory-motor capabilities of computer systems to better match the natural communication means of human beings. The purpose is to enhance interaction between the user and the computer by using appropriate modalities to improve:

- the information bandwidth between the human and the computer; that is the amount of information being communicated,
- the signal-to-noise ratio of conveyed information; that is the rate of information useful for the task being performed (Maybury, 1993).

Although the potential for innovation is high, the current understanding of how to design and build multimodal user interfaces is still primitive. The power and versatility of multimodal interfaces result in an increased complexity that current design methods and tools do not address appropriately. This problem is exacerbated by the proliferation of new input and output modalities resulting from the combination of modalities. Indeed multimodal user interfaces support interaction modalities, which may be used sequentially or concurrently, and independently or combined synergistically (Nigay & Coutaz, 1993). Novel aspects of interactions must be considered, such as the fusion and fission of information, and the nature of temporal constraints on the interactions (Bellik, 1997).

In this article we address this problem of design and implementation of multimodal user interfaces by presenting a design process for multimodal interaction. The structure of the article is as follows: first we present the phases of our design process. We then present the tools dedicated to support some phases of our design space. In particular we focus on the tools for developing multimodal interfaces and the resulting software architecture of a multimodal interface.

2 The INTUITION design process

The main goal of INTUITION is to provide a sound and tool-supported methodological framework enabling to introduce multi-modal interaction into the highly safety-critical domain of military fighter cockpits. To this end, we advocate a design process which is iterative and model-based at the same time (Figure 1).

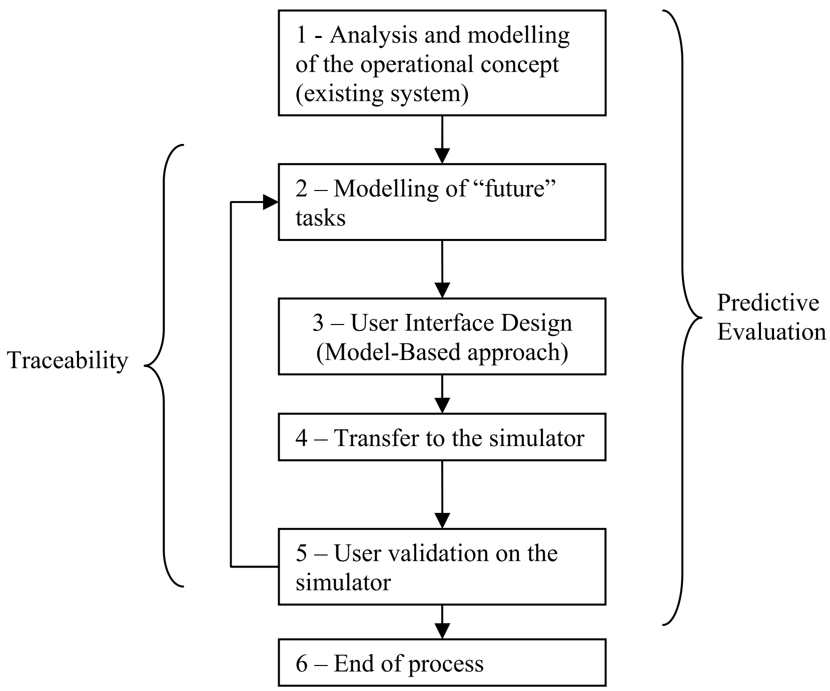

Figure 1: The six steps of the INTUITION design process

Phase 1: The first step of the design process is to perform a detailed analysis of the existing system. The goal of this analysis is to investigate which amongst the user’s tasks would benefit from the increased human-computer bandwidth or signal-to-noise ratio that multimodal interaction can provide.

Phase 2: having identified one or several tasks that are candidate for multimodal support, we perform a detailed task modeling in the framework of the envisioned system. The result of this analysis will be used as a reference for all design activities in phase 3, and also as the basis of user validation in phase 5.

Phase 3: Following this first tasks analysis step, our process moves on to a User Interface Design phase, relying on a model-based approach. Several models, along with their supporting tools, are used during this phase (these models and tools are further described in §3.)

- Architectural models: INTUITION relies on the ARCH (Arch, 1992) and MVC architectural models, which mandate a clear functional decomposition of the user interface.
- Multimodal input: The multimodal interaction is described using the ICARE approach (Bouchet & al., 2005), which also provides the benefit of code generation.
- Multimodal output: The sensory channels used for providing information can be elected **at** runtime, depending on contextual information. For instance, 3D sound can be preferred over graphic output to prevent information overload. Multimodal output is described in the MOST framework (Rousseau & al., 2005).
- Dialogue structure: in our application domain, the dialogue between user and system is usually very concurrent and time constrained. This kind of dialogue is challenging to specify, and to this end we have selected a suitable formal notation, Petri nets. Petri nets have the advantage of being executable (which is a requirement for step 5, user validation) and are also amenable to mathematical analysis.

Phase 4 and 5: The INTUITION design process is user-centered, encouraging the building of prototypes that can be evaluated inside a full-scale aircraft simulator by end users (in our case, military pilots). The models used in phase 2 have in common to be executable: They can be transferred to the simulator (phase 3) and executed in this simulated environment. This enables us to follow an iterative process, where pilots can evaluate the newly designed interaction technique. This experimental evaluation is done during phase 4 (User validation). The pilot is required to perform some pre-defined tasks, and we perform both quantitative (e.g. time for task performance, number of errors) and qualitative (user satisfaction, comments and suggestions) assessment for the current iteration of the design. The interactive nature of our tools enables us to change in a large way several aspects of the interaction technique at run-time, thus making easy to evaluate several distinct interaction scenarios during one simulation session.

According to the results of the evaluation phase, we may perform new iterations of the cycle in order to improve the design of the interaction technique, or terminate the process.

Two concurrent activities are carried out in parallel with the iterative design process:

- Predictive Evaluation: The models used in phase 3 support different forms of predictive evaluation. For instance, Petri nets used for dialogue modeling can provide predictive information on task performance, in terms of number of elementary actions or of minimal time required to perform the task. ICARE provides predictive evaluation of multimodal properties (Complementarity Assignment Redundancy Equivalence). MOST can detect inconsistency in the modalities selection rules.
- Traceability: It is essential that the output of the INTUITION process consist not only of a validated design, but also of a rationale for this design. We need to remember and to trace the design decisions that have been made throughout the process, what other options have been considered, and the reasons why these options have not been retained. This activity is supported by a Design Rationale notation based on QOC (Questions, Options, Criteria), and its support tool, DREAM.

In the following section, we focus on the tools dedicated to the development phase.

3 Tools for developing multimodal user interfaces

Although multimodal interfaces provide a powerful style of interaction, they are functionally more demanding and therefore more complex to develop than, for example, graphical user interfaces. As a consequence**,** the development of multimodal interfaces is a difficult task. Development tools dedicated to multimodal interaction are currently few and limited in scope. Either they address a very specific technical problem such as media synchronization (Little et al., 1991) or they are dedicated to specific modalities such as speech or gesture. As opposed to these existing tools, in the INTUITION project, our goal is to design and develop a general tool for developing multimodal interfaces without being dependent on particular modalities.

A development tool is based on an underlying software architecture model that structures the code of the multimodal user interfaces developed using the tool. A software architecture is an organization of computational elements and the description of their interactions. An architectural model such as pipes and filters, object-oriented organizations and layered structures, can be used as generic vehicles to express architectural solutions. An architectural model makes recurring organizational patterns explicit (Garlan & Shaw, 1993). A model defines a vocabulary of design elements, imposes configuration constraints on these elements, determines a semantic interpretation that gives meaning to the system description, and enables analysis of the system properties. As a result, architectural models not only lead to different design solutions but to designs with significantly different software properties (Shaw & Garlan, 1995).

In the domain of user interface, a number of architectural models have emerged. Seeheim (Pfaff, 1985), MVC (Goldberg, 1984), PAC (Coutaz, 1987), Arch (Arch, 1992) and others can be considered as canonical references. They primarily provide a framework for performing functional partitioning and allocation of function to structural components based on system and user-centered properties. One architectural model, PAC-Amodeus (Nigay & Coutaz, 1995) is dedicated to multimodal user interfaces. PAC-Amodeus uses the Arch model (Arch, 1992) as the foundation for the functional partitioning of an interactive system and populates the key element of this organization, i.e., the Dialogue Controller, with PAC agents.

From PAC-Amodeus, we reuse the modality dependent and independent partitioning based on Arch. As show in Figure 2, we structure a multimodal system according to the Arch model and we distinguish the modality dependent components from the modality independent components. Arch incorporates two adaptor components, the Functional Core Adaptor and the Logical Interaction Component, to insulate the keystone component (i.e., the Dialog Controller) from modifications in its unavoidable neighbors: the Functional Core and the Low Level Interaction Component. As a result, modifying the interaction modalities and their combined usages (forms of multimodality) will imply to only change the two components Low Level Interaction and Logical Interaction, the Dialog Controller being not modified. Since the concepts manipulated by the Dialog Controller are modality independent and concern the task sequencing, while the two other components focus on modalities and multimodality we decided to use a specific tool dedicated to specify and develop the Dialog Controller. The INTUITION tool is called PetShop and enables the developer to specify the Dialog Controller as a Petri Net. PetShop is fully described in (Bastide & al., 2005).

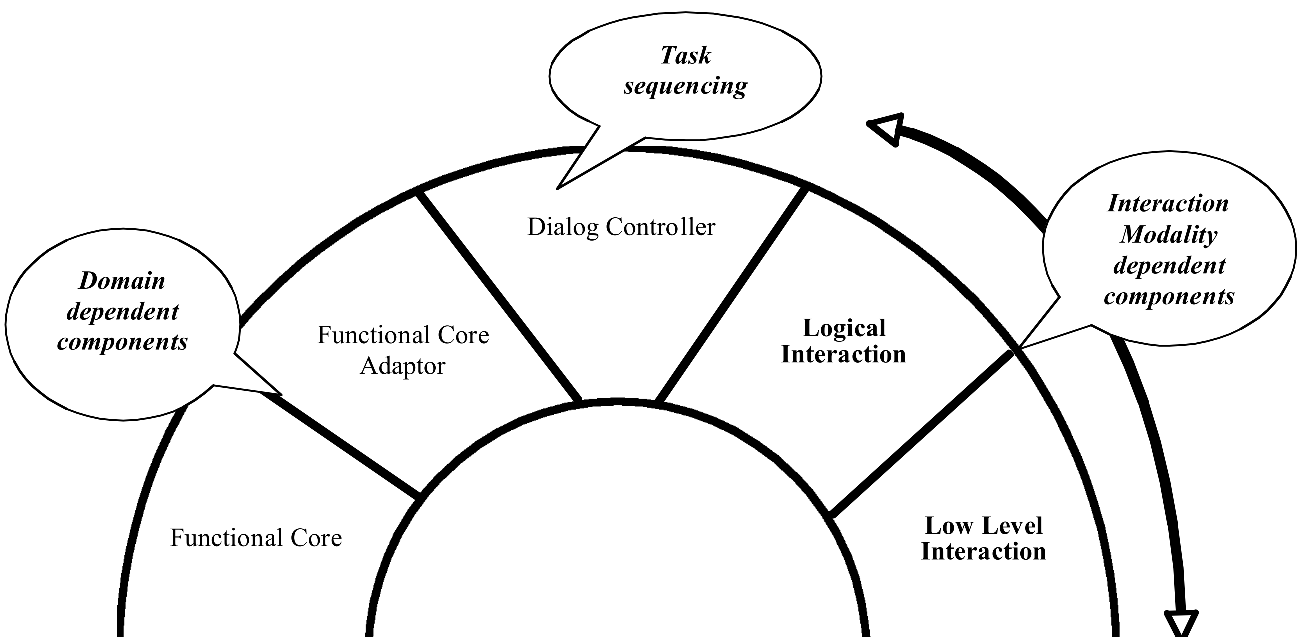

Figure 2: Arch model: Modality dependent/independent components. Two components, namely Logical Interaction and Low Level Interaction, are dependent of the interaction modalities. The three other components are modality independent.

For developing the multimodal interaction components, we propose two tools. Indeed we decided to distinguish the input multimodal interaction from the output multimodal interaction because the characteristics of input and output multimodality are different. For inputs, one of the main salient features is the interpretation function from low-level interaction events to a command or elementary task understandable by the Dialog Controller. This interpretation function involves fusion of data along different modalities in order to obtain a complete command. For outputs, one of the main challenges is the relevant choice of modalities and forms of multimodality for rendering information to the user: the presentation function. This function involves selecting the modalities and forms of multimodality based on the interaction context including the user's profile, the system resources and the physical environment in order to present information to the user in an adequate way without disturbing or surprising her/him.

By distinguishing input and output interfaces we reuse the MVC decomposition, which advocates one component dedicated to the input user interface, namely the Controller, and one component to the output user interface, namely the View. Nevertheless it is important to note that MVC is an agent-based model. The MVC model therefore structures an interactive system as a collection of specialized computational units called agents. A MVC agent is modeled along three functional perspectives: the Model, the View, and the Controller. A Model defines the abstract competence of the agent (i.e., its functional core). The View defines the perceivable behavior of the agent for outputs. The Controller denotes the perceivable behavior of the agent for inputs. The View and the Controller cover the user interface of the agent, that is, its overall perceivable behavior with regard to the user. As opposed to MVC, which is an agent-based model, we use the functional partitioning of Arch and we reuse from MVC the distinction between input and output interfaces.

The INTUITION tool dedicated to input multimodal user interfaces is called ICARE, 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. ICARE is fully described in (Bouchet & al., 2005). The inputs of the ICARE-generated code are the actions performed by the user along several modalities. The outputs of the ICARE-generated code are elementary tasks that are modality independent and transferred to the Dialog Controller.

The INTUITION tool dedicated to output multimodal user interfaces is called MOST, for Multimodal Output Specification Tool. MOST includes an election module that performs the right choice of modalities and forms of multimodality (complementarity, redundancy) for presenting information to the user. The election module relies on three types of knowledge: interaction context (as models), application behavior (as rules) and a Multimodal Presentations List (MPL). MOST is fully described in (Rousseau, Bellik, Vernier & Balzagette, 2005). The inputs for MOST are the semantic information to be conveyed to the user and provided by the Dialog Controller. The outputs of MOST are the descriptions (in the MOXML language -Multimodal Output extended Markup Language) of the multimodal presentations that are then realized along different modalities.

In summary, three INTUITION tools are dedicated to the development phase. They are presented in Figure 2 in the context of FACET, a real-time simulator of the French military aircraft RAFALE.

- One tool (PetShop) is dedicated to the Dialog Controller and focuses on task sequencing.
- One tool (ICARE) is dedicated to multimodal input user interfaces and includes the fusion of data specified by the user along different modalities.
- One tool (MOST) is dedicated to multimodal output user interfaces and includes a process for selecting the output modalities and forms of output multimodality for rendering information to the user.

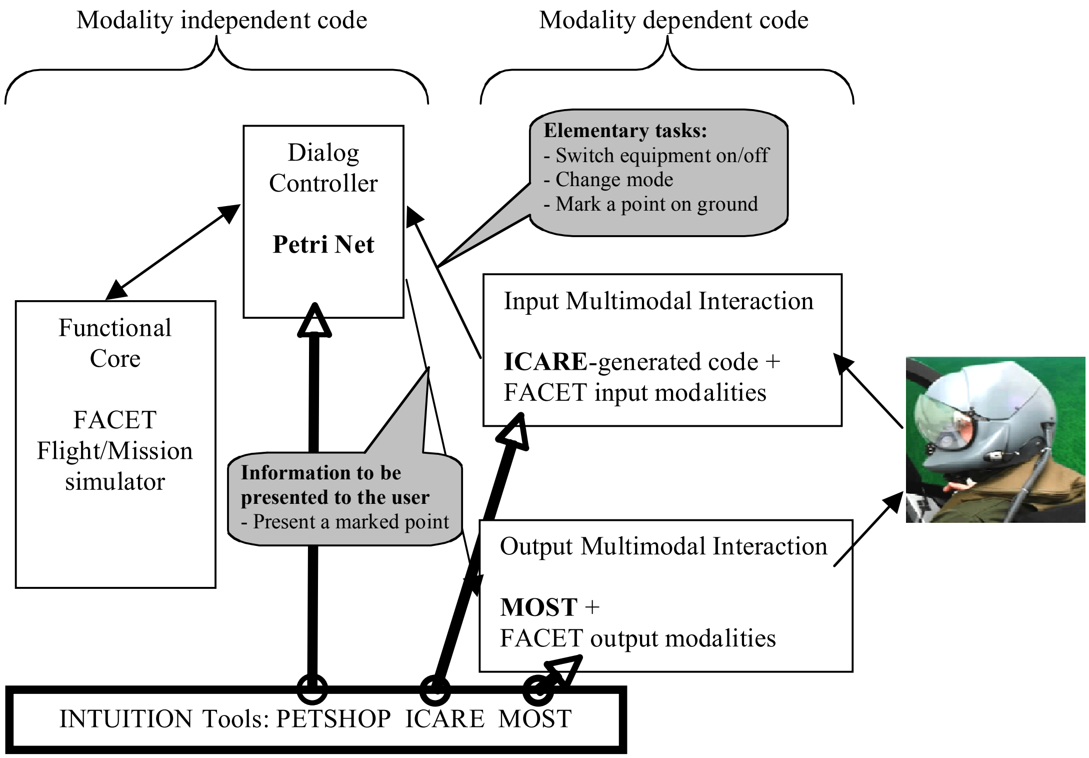

Figure 2: Multimodal interaction for military aircraft cockpit: Three INTUITION tools for developing the multimodal interface

4 Summary and future work

In this article, we have presented our design process for multimodal interaction. We have then exposed the tools that we developed for supporting some phases of our design process. In particular we have focused on the development phase of the design process by presenting our development tools. We have also explained the resulting MVC-like software architecture of a multimodal interface developed using our tools.

As part of the INTUITION project, we have applied our design process for designing and developing multimodal interaction for military aircraft cockpits in FACET (a real-time simulator of the French military aircraft RAFALE). This has shown the applicability of our design process. In addition for the development phase, we also validated our development tools and resulting software architecture for a real-time system. In the INTUITION project, we will continue to test and verify our results through the design of systems supporting multiple modalities. Our next step is to apply our design process for other tasks supported by FACET as well as for another application domain, namely Air Traffic Control, for which we would like to explore the flexibility offered by our development tools.

5 Acknowledgments

The work presented in the article is funded by DGA (French Army Research Dept.) under contract INTUITION #00.70.624.00.470.75.96.

6 References

Arch, A Metamodel for the Runtime Architecture of an Interactive System. The UIMS Tool Developers Workshop. *SIGCHI Bulletin*, 24, 1, 1992, (pp. 32-37). ACM Press.

Bastide, R., Barboni, E., Balzalgette, 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.

Bastide, R., Lacaze, X., Navarre, D., Palanque, P., & Galindo, M. (2005). Can we rationalize the design and construction of Air Traffic Management system. In *Conference Proceedings of HCI International 2005, the 11th International Conference on Human-Computer Interaction*, Las Vegas, Nevada, USA, July 2005. Lawrence Erlbaum Associates.

Bellik, Y., Media Integration in Multimodal Interfaces, In *Proceedings of the First IEEE Workshop on Multimedia Signal Processing*, Princeton, New Jersey, June 1997.

Bouchet, J., Nigay, L., & Ganille, T. (2005). The ICARE Component-Based Approach for Multimodal Input Interaction: Application to real-time 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.

Coutaz, J. PAC, an Object Oriented Model for Dialog Design. In *Conference Proceedings of Interact'87*, Stuttgart, Germany, Sept. 1987, (pp. 431-436). North Holland.

Garlan, D. & Shaw, M. An Introduction to Software Architecture. In *Advances in Software Engineering and Knowledge Engineering*, V. Ambriola and G. Tortora Eds., Vol. 1, 1993, (pp. 1-39). World Scientific Publ.

Goldberg, A. *Smalltalk-80: The Interactive Programming Environment*, 1984. Addison-Wesley.

Little, T.D.C, Ghafoor, A., Chen, C.Y.R., Chang, C.S., Berra, P.B. Multimedia Synchronization. In *IEEE Data Engineering Bulletin*, Vol. 14, 3, Sept. 1991 (pp. 26-35).

Maybury, M.T. (Ed.). *Intelligent Multimedia Interfaces*, 1993. AAAI Press/MIT Press.

Nigay, L., & Coutaz, J. 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.

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

Pfaff, G.E. et al. User Interface Management Systems. *Eurographics Seminars*, 1985. Springer Verlag.

Rousseau, C., Bellik, Y., Vernier, F., & Balzagette, 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.

Shaw, M. & Garlan, D. *Software Architecture. Perspectives on an Emerging Discipline*, 1995. Prentice Hall.