Architecture Framework For
Output Multimodal Systems Design

Cyril Rousseau[1]
Dr Yacine Bellik[1]
Dr Frederic Vernier[1]
Dr Didier Bazalgette[2]
(1) Paris XI University
(2) French Ministry of Defence

LIMSI-CNRS
Paris XI University
Orsay, France
Email: cyril.rousseau@limsi.fr
yacine.bellik@limsi.fr
frederic.vernier@limsi.fr

General Delegation for the Armament
Ministry of Defence
Paris, France
Email: didier.bazalgette@dga.defense.gouv.fr

Abstract

An output multimodal system aims at presenting information in an "intelligent" way by exploiting different communication modalities. According to the desired multimodal system, this notion of "intelligence" may vary. However all existing systems share the same goal: the information presentation must be the most suitable to the interaction context. In this study we present a software architecture model for dynamic and contextual Human-Computer Interaction systems. Our proposed architecture framework is more precisely suited to the output side of multimodal systems and introduce certain mechanisms, which are not available with classical GUI architectures, to tailor the information expression to the interaction context. Two Applications of this architecture framework (mobile telephony and military avionics) are also described.

Keywords

Human-Computer Interaction, output multimodality, multimodal system design, interaction context.

INTRODUCTION

Future information processing systems will touch more and more diverse categories of users (beginners, children, elderly people, etc.). Nowadays, the use of computer is taught from the earliest age and the use of computers in the school will emphasize this trend during next decade. Likewise, the adaptation of computer hardware to the disabled persons is now expanding very rapidly and concerns as well visually challenged people (Bellik and Farcy 2002), as deaf people (Braffort et al. 2004), as people with autism (Grynszpan et al. 2003), etc. Consequently, future systems tend to be able to adapt to all kind of users.

In the other sides, computer markets are more and more diverse: workstation, notebook computer, PDA (Personal Digital Assistants), mobile phone, etc. Software is generally specific to the development platform. Extension to other support requires new developments and costs money. In consequence, new systems tend to be more "plastic" (Thevenin 2001): in other words, they are able to adapt to different platforms. This will afford to limit the production costs and to bring some uniformity.

Lastly, we can see information processing systems are now present in new places such as cars, trains, planes, etc. The computer is not considered anymore as a working tool only but also as a means of entertainment, service and discovery. The size, the weight and the cost of notebook computers have decreased and wireless local network areas have appeared. These changes make working or relaxing with a computer in all kind of places (like parks, stations, airports, pubs, fast foods, etc.) possible now. This is why future systems must be able to adapt to different environments of use.

In conclusion the best word to characterize future information processing systems is "diversity". This diversity is expressed at different levels: diversity of users, diversity of platforms and diversity of use environments. To deal with these diversities, it will be necessary to build systems able to adapt to this diversity.

A certain number of concepts relating to this idea of diversity have been established by different authors (Stephanidis and Savidis 2001): the concept "Universal Access", the concept "Dual User Interfaces", the concept "User Interface for All" (UI4ALL), the "plasticity" concept (Thevenin 2001), and many more besides. All these concepts are focused on the same idea: the definition of a "universal" system, more precisely an adaptable and adaptive system. On the one hand, it is adaptable in the meaning that the system can be configured by the user according to his preferences. On the other hand, it is adaptive in the meaning that the system is able to detect the modifications of the interaction context and to adapt consequently.

The work presented in this article, has been implemented within the framework of the INTUITION project (multimodal interaction integrating innovative technologies) partly funded by French DGA (General Delegation for the Armament) and including three French laboratories (LIMSI, CLIPS-IMAG and IRIT) and an industrial partner (THALES-Avionics). The objective is to develop an adaptation platform of the new technologies in Human-Computer Interaction.

Our study is built on three steps. First, we present our definition of the output multimodality concept and underline the relevant contributions. Thereafter, we propose an architecture model for a dynamic and contextual Human-Computer Interaction. Lastly, we describe two applications developed in order to validate our model.

OUTPUT MULTIMODALITY

In our point of view, output multimodality consists in expressing and presenting information to the user in an "intelligent" way by exploiting different communication modalities. Some authors use the MIIP (Multimodal Intelligent Information Presentation) term to make reference to this concept.

The first output multimodal systems like CUBRICON (Neal and Shapiro 1991) or INTEGRATED INTERFACES (Arens et al. 1988) had as main objective the reproduction of the natural communication between a user and his computer. The authors of these systems defined the basis of the output multimodality domain and the different tasks to realize, like the coordination of two modalities of the same mode (WIP (André et al. 1993) and COMET (Feiner and McKeown 1993)) or of two different modes (MAGIC (Dalal et al.)) (Figure 1 (Rousseau 2003)). Towards the end of the Nineties, systems like VICO (Bernsen and Dybkjaer 2001) and particularly AVANTI (Stephanidis and Savidis 2001) dealt with the problem of the effect of the interaction context on the Human-Computer Interaction.
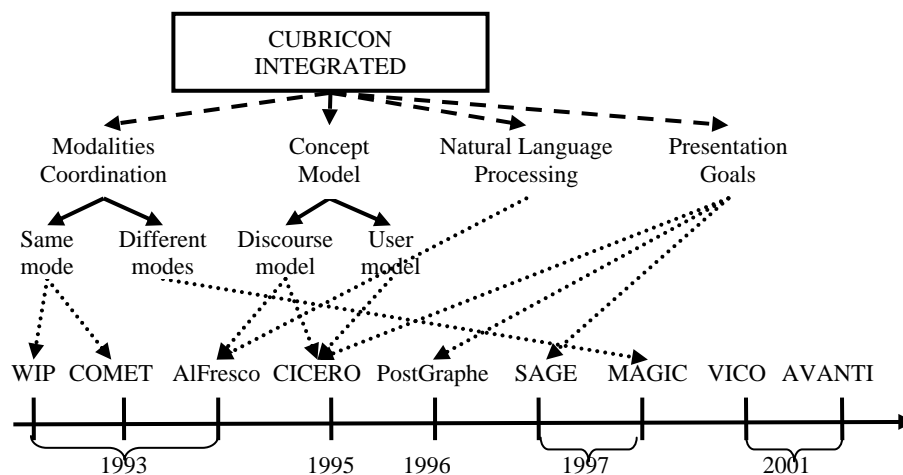
Figure 1: Repercussion of the concepts introduced by the first multimodal systems.

This study of output multimodality domain revealed a very great diversity of existing output multimodal systems. VICO is a driver assistance system, MAGIC generates medical briefings (postoperative status), WIP is an automatic generator of manuals, etc. Objectives and application fields are often different from a system to another. These systems depend on application field and interaction context for which they were initially developed. Some systems are targeted on user, others on task, others on system or environment and so these dependences limit the extensibility and the reutilisability of these systems.

If we take for example the mobile telephony field, next generation of mobile phones will have a GPS chip. This new media will allow localizing user and so improving Human Computer Interaction. For example we can detect a user's entry in a cinema and switches the user's phone in silent mode. This requires an environment representation but the environment concept is a new idea in mobile telephony field. This will probably require the definition of a new phone system in order to manage the environment.

In the same way, our industrial partner developed a fighter simulator in which the pilot's HMV (Helmet Mounted Visor) is disabled when the pilot looks inside his cockpit. The only way to modify this behavior is to change the simulator code. This simple modification becomes a tiresome task. The representation in a declarative way of the application behavior and of the interaction context will improve the reusability and the application extensibility.

These considerations highlight a need to define a generic architecture model. This architecture must have a representation of application behavior and interaction context which gathers information on the task, the user, the system and the environment. Then, starting from this context representation, the system should be able to detect in real time the modifications in this context and change its behavior consequently in a dynamic way, i.e. during application execution.

INTERACTION CONTEXT

Consequently, one of the main characteristics of an output multimodal system is its capacity to detect modifications of the interaction context and to make right decisions without disturbing or surprising the user. We are talking about adaptive system or about "context aware" (Dey 2001). This property can only be guaranteed if the system has enough information about the interaction context. That is why the interaction context has to be specified and updated in a regular way. Information that characterizes this context is generally stored in adapted structures called "models" (user model, system model, task model, etc.).

A model is defined by a set of criteria which breaks up into two parts: the first one is static and the other one is dynamic. The static part contains pre-established knowledge about the model. The dynamic part refers to knowledge which is likely to evolve and which is generally stored in the form of logs. For example, in the case of the "user" model, the user language and the direction of the glance are respectively criteria of the static and dynamic parts of the user model.

Available models and knowledge that are taken into account within these models may be different according to studied systems. Some systems draw up precise definitions about the user model or the system model (Arens and Hovy 1995). But these definitions may be not necessarily useful for other applications.

Therefore, the definition of the different elements of the context, which have to be taken into account, must be customizable (adaptable) by the designer. An element of the context can be relevant for some kind of application field and useless for other ones. For this reason, we can say that the used models and knowledge are not the same from a multimodal system to another. Many researches are trying to define an ontology relating to interaction context (Gu et al. 2004) but these research works are not yet advanced enough to be really exploited. The approach that we present at the following section corresponds to a modular architecture which should finally enable us to incorporate the works made on the context, without requiring structural changes of our proposal.
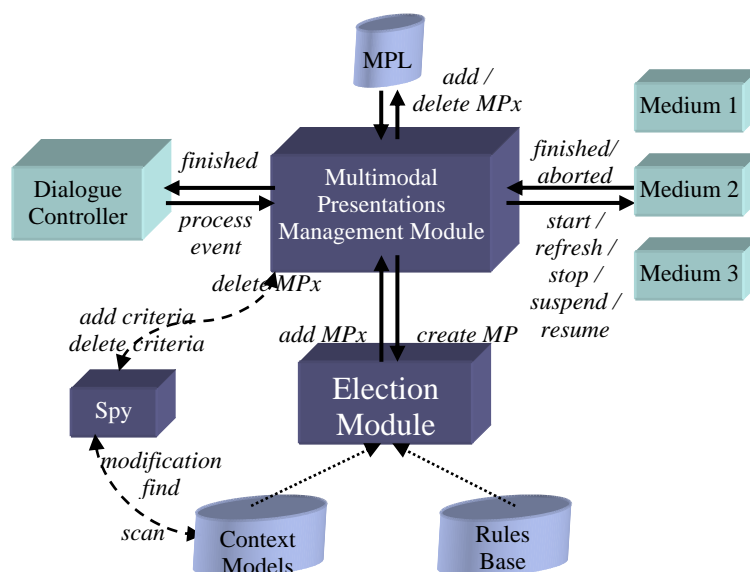
SOFTWARE ARCHITECTURE

Figure 2: Software architecture.

The concepts presented previously inspired our proposition for a generic software architecture of an output multimodal system. Figure 2 is a part of the global architecture suggested within the framework of the INTUITION project, which deals with the system outputs. We called the system resulting from this architecture MOST, for Multimodal Output Specification Tool. A data representation language MOXML (Multimodal Output eXtended Markup Language) (Rousseau 2003) is associated with this tool. This language allows to describe in a descriptive way all the elements required by the architecture (modes[1], modalities[2], media[3], relevant elements of the interaction context, election rules, etc.).

Election Concept

The architecture presented in Figure 2 is based on a concept which we call "election". The election starts from a semantic information decomposed into Elementary Information Units[4] (EIU) and then applies certain rules to present it to the user in a pertinent way. This presentation consists in allocating for each EIU a multimodal presentation, more precisely a combination of output modality-medium pairs. The election system elects for each EIU the best output modality-medium pairs set to express it.

Some authors use the word "fission" by the opposite to the word "fusion" to name the selection process of output modalities. We think this term is not relevant for this use.

Indeed, there is a fission, more precisely a decomposition but this fission takes place on the semantic level. So we prefer to talk about fission during the decomposition of the semantic information in elementary information units (EIU) and of "election" when affecting modalities to elementary information units. We chose the term "election" by analogy with a voting system of a political election. Indeed our election process is based on a rules base (voters) which once applied add or remove points (votes) to certain modes, modalities or media (candidates), according to the running state of the interaction context.

Architecture Description

The architecture is composed by three main modules: the election module, the MPMM (Multimodal Presentations Management Module) and the Spy. Concerning knowledge, it is distributed in three different structures: models (interaction context), a rules base (application behavior) and the MPL (Multimodal Presentations List). The dialogue controller (on the left of Figure 2) has a central position regarding to the global architecture of INTUITION project. It allows communication between MOST and the other elements of the platform. It communicates with the MPMM through messages. The MPMM treats the messages coming from the dialogue controller by generating a multimodal presentation allowing the expression of the semantic contents associated with the message. Once the multimodal presentation allocation finished, this presentation is transmitted to the different media (on the right of Figure 2).

The election module can be considered as the heart of the architecture. This module builds the multimodal presentations. It uses a set of knowledge distributed in two structures types: models and rules. Information relating to the interaction context is stored on several models and a rules base defines the election behavior. The election module applies the rules base which modifies the contextual weights associated to the interaction components (modes, modalities, media) managed by the system. The values of these contextual weights depend of course on the interaction context and will decide which multimodal presentation will be elected. Once the application of the rules base carried out, the best modality-medium pair is selected. The system will then make the choice to "enrich" or not the multimodal presentation. "Enrich" means to select new modality-medium pairs regarding CARE (Coutaz et al. 1995) properties (redundant or complementary pairs to the first one). Finally, the election module sends to the MPMM the elected multimodal presentation.

The MPMM centralizes the architecture communications. It is a mediator module which distributes work between the different modules of MOST. Nevertheless, its role is not limited to centralize the communications. It aims also at managing the list of the active multimodal presentations. To do that, it uses the MPL (Multimodal Presentations List) which refers to the active multimodal presentations as well as the elements of the interaction context which make these presentations valid. Actually the MPMM is also responsible to check the validity of the active multimodal presentations. It receives information from a spy module, charged to analyze the evolution of the interaction context. This spy announces any modifications of an element of the context having an influence on an active multimodal presentation. The MPMM may then decide to cancel the validity of some presentations. In the case of a cancellation of an active multimodal presentation, the MPMM asks the election

---

[1] An output mode corresponds to a user's sensory system (visual, auditory, etc.).

[2] An output modality is an information structure perceived by user (text, graphic, sound, etc.).

[3] An output medium is an output device allowing the expression of an output modality (screen, speaker, etc.).

[4] An elementary information unit (EIU) is an atomic unit which can be expressed by a modality.

module to launch a new election. The election module will then return a multimodal presentation associated with a criteria list of the interaction context in order to assure the validity of the new presentation.

Lastly, the Spy module has a key role like it was said previously. It may be the source of possible invalidations of certain active multimodal presentations. However the operating principle of this module is rather simple. As soon as a multimodal presentation is elected (and added to the MPL), the Spy receives a criteria list of the interaction context on which this presentation depends. Then, it supervises the modifications of these criteria. The Spy module announces to the MPMM all modifications which may cancel the presentation validity and thus requiring a new election.

Multimodal Presentations Instantiation

We carried out a distinction between the modalities choice and their instantiation. As we have just seen it, the modalities choice is done by the election module. It remains to determine how to present the information through these modalities. This step called "multimodal presentation instantiation" consists in generating the multimodal presentation allocated by the election engine. Finally we can say that election is "*Which* modalities to present the information?" whereas instantiation is "*How* to present the information using these modalities?".

This generation breaks up into two steps. The first step (called "content layer") consists in choosing the content to express through the modalities making up the allocated multimodal presentation. The second step (called "realization layer") makes choices on the presentation parameters (position, modalities attributes, etc.).

For example, an "intelligent" mobile phone wishes to announce a call reception. The election module expresses this information through a multimodal presentation composed of: "Text" modality with "Screen" medium and (a redundant pair) "Ringing" modality with "Speaker" medium. The "content layer" chooses for example "Call of …" for the first modality and the pink panther song for the second. The "realization layer" chooses for example to put text in the center of the screen with a significant character font for the first modality and to turn the volume up for the second.

The multimodal presentations instantiation is currently handled by the media. In the case of the INTUITION project, a Thales-Avionics output generation engine (not visible on the figure 2) is associated with each medium and manages this phase. However, instantiation is a subject of research in full rise (Rist and Brandmeier 2002) and remains a research prospect for middle-term.

Data Representation Language

The first objective of our works was the definition of a generic output multimodal system. For that, we decided to separate knowledge and engine. A data representation language (MOXML) was defined in order to represent the different knowledge needed to specify system outputs.

MOXML is based on XML (eXtensible Markup Language) which appeared really suitable to our needs. Available tags can be defined by the user and this property confers to it the possibility of using it in any application field. MOXML defines a set of tags which allows to describe all the needed elements in an output multimodal system.

Here is an example of an election rule described in MOXML within the framework of the application "Ground marking out in a fighter plane" (presented in the next section). This rule means that if the pilot lowers his head when the system needs to present operation results, the system will not use HMV (Helmet Mounted Visor) medium in order to improve the cockpit screens visibility.

```
<rule name="Pilot's head position low" number="1">
   <premises>
      <premise number="1">
         <elt_left model="user" criteria="head position" />
         <operation name="=" />
         <elt_right type="string" value="low" />
      </premise>
   </premises>
   <conclusions>
      <conclusion number="1">
         <target level="media" name="Helmet Mounted Visor" />
         <effect value="forbidden" />
      </conclusion>
   </conclusions>
</rule>
```

Rules Base Design Process

We saw through the last sections the utility, the use and finally the representation of the election rules. However, we did not explain how to build this rules base. A design process of the rules base (Figure 3) has been developed in collaboration with Thales-Avionics, our industrial partner.
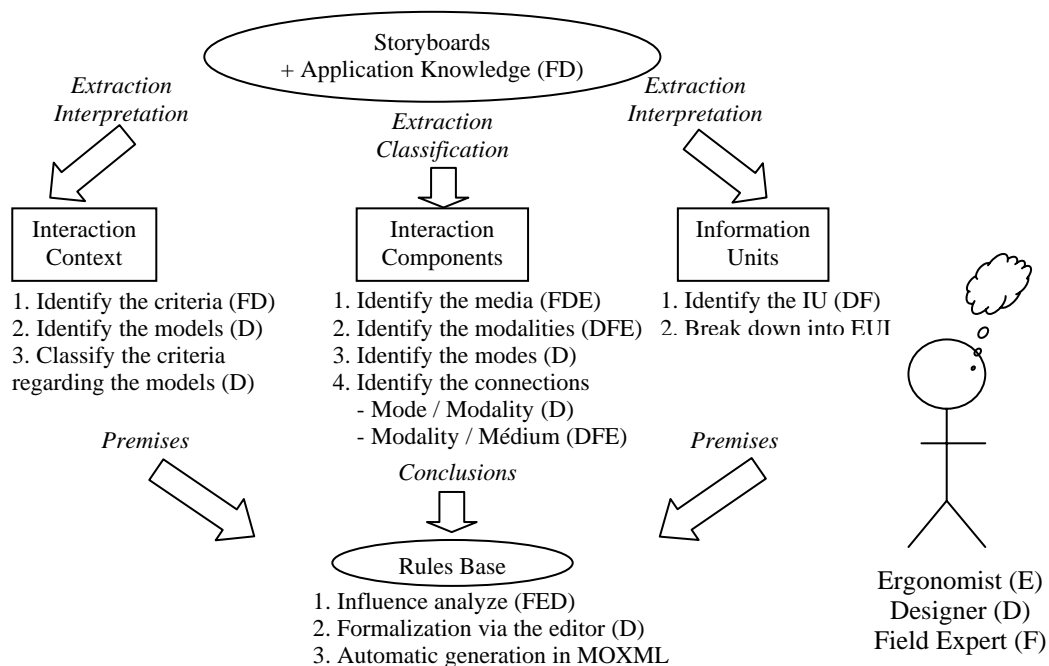

Figure 3: Rules Base Design Process.

This process is based on a data corpus, which gives the elementary elements needed to build the rules base. It is required the participation and collaboration of three actors: an ergonomist actor, a designer actor and a user actor (expert in the application field). The choice of the corpus is particularly important. The quality of system outputs will highly depend on this choice. The corpus must be composed of a consequent and diversified set of scenarios / storyboards (referring to nominal or degraded situations) but also of relevant knowledge on field application, system, environment, etc.

The interaction context modeling (left branch of the Figure 3) consists in identifying through the chosen scenarios, the pertinent data relating to the context which can influence the interaction. These data are then interpreted by the actors in order to constitute criteria and then classified by categories called models. The central branch of the Figure 3 consists in identifying the interaction components (modes, modalities and media) which should be managed by the system. In general, the interaction components are not hard to extract from scenarios. Media are often defined in the technical documentations and from media it is relatively easy to specify modes and modalities. Finally, the right branch specifies the Information Units (IU). Once the IU list defined, these one are broken down into Elementary Information Units (EIU).

Once these required elements defined, the actors can start to specify premises and conclusions rules. A graphical rules editor allows to edit the rules base in a transparent way without using an XML editor. Moreover the editor will automatically and incrementally verify the structural coherence and the structural fullness of the rules base.

SOFTWARE VALIDATION

Two applications have been developed in order to validate the software architecture. The first application "Call Reception on a mobile phone" will be briefly presented. We will give more details about the second application "Ground marking out in a fighter plane" which exploits all the architecture modules. Lastly we will discuss the applications contribution.

Call Reception on a mobile phone

A first application of this software architecture has been built in order to validate the election module. The choice of this application was based on the mobile telephony field. It aims at the simulation of a phone call reception on an "intelligent" mobile phone. In other words, this application announces in a dynamic and contextual way, the reception and the origin of a call to its owner.

The mobile phone has for main drawback to be intrusive. The user can be contacted at any time and sometimes this is turned over against its user. Any mobile phone user knew and still knows this unpleasant situation which consists in receiving a call at an inappropriate moment or in an inappropriate place. The solution to this problem consists usually in turning off the phone or in enabling the silent mode. This underlines the problem of current mobile telephony: the phone is customizable thus adaptable but in no case adaptive. The mobile phone "intelligence" beside the context is currently null. However, the mobile phone has the capacity to present a new incoming call in several ways. It has several output devices such as screen, vibrator, speaker and diodes. Thus, the mobile phone seemed to us to be a perfect subject of experimentation and validation of our proposal.

In the developed application, according to the specified interaction context (noise level, user place, phone position, battery level, etc.), the phone makes choices based on performance and comfort criteria. If the phone battery level is low, it will try to save remaining energy by minimizing for example the use of the screen. In the same way, if the person goes to a noisy place like a pub and receives a phone call, the phone will try to present it by using for example vibrations. In this application, we did not want to limit us to currently "measurable" context elements. These criteria like the user place (hospital, cinema, pub, station, etc.) are not unrealistic. Indeed the next generation of mobile phone will have for example a GPS chip making it possible to locate the user inside buildings.

The application behavior is specified in a descriptive way through the rules used by the election module. Therefore, it is easy to modify this one and to adapt it by modifying or by enriching the rules base. An editor and checking mechanisms (to ensure that the rules are structurally complete and good) make this task easier. For this application, 15 interaction components including 3 modes (visual, auditory and tactile), 7 modalities (text, photography, logo, color, ringing, synthetical voice and vibration) and 5 media (screen, diode, vibrator, speaker and earphone) are managed by the system. The interaction context is defined through 28 context criteria (battery level, media availability, phone position, user preferences, etc.). Lastly, the election behavior was specified through 61 rules.

Ground marking out in a fighter plane

The generic platform of the INTUITION project in which this work takes place must be usable in real time. This constraint is critical in the context of a fighter plane, where the shortest delay may lead to dramatic consequences. Thus it will be necessary to optimize the multimodal presentations election phase in order to allow real-time responses.

This is why an application was carried out within the framework of the INTUITION project in order to check the real time constraint but also to validate the MPMM of the software architecture, which has a more prevalent role in this second application. The application subject is about a task of marking out a target on the ground realized in a fighter plane cockpit. It is related to a RAFALE (French fighter) simulator, developed by Thales-Avionics.

Apart from the validation aspect, this application has also an improvement interest of the pilot's performances. The most important factor in the military avionic field is about the pilot's reaction time. He reacts more or less quickly according to his speed to perceive and interpret the information transmitted by the system. This is why, the choice made regarding the information presentation is very important. It may improve the computer-human communication in order to make perception and comprehension times shorter, and thus to improve pilot's reactivity times, and indirectly his survival chances.

Design Process Illustration

We will now illustrate the rules base design process (Figure 3, subsection 4.5). For that, we will present the different process steps within the framework of this application.

At first, the process has been applied by three actors: a designer and two avionic experts. It was not possible to have an ergonomist for this application. However the presence of a second expert in the application field was particularly useful.

In the avionic field, the corpus refers for example to knowledge on the plane (technical documents, plane HCI, etc.), on the crew (number of people, tasks of each one, etc.), on the regulation (civil and military procedures), on the environment (meteorology, etc.), on the personnel type (work environment, human performances, etc.), etc. It underlines the complexity and the quantity of work to do in order to get a complete and relevant corpus, two properties which will guarantee the generation of a good rules base.

In the other sides, the interaction context is specified through 8 criteria: pilot's head position, NAS (Navigation and Armament System) mode, media availabilities, audio channel availability, luminosity level and noise level. The criteria classification uses three models: user model (pilot's head position), system model (availabilities)

and environment model (levels). In this case, the classification is relatively easy but in other case this step can be more difficult. Sometimes, some criteria can be classified in two models. It is also necessary to make a choice.
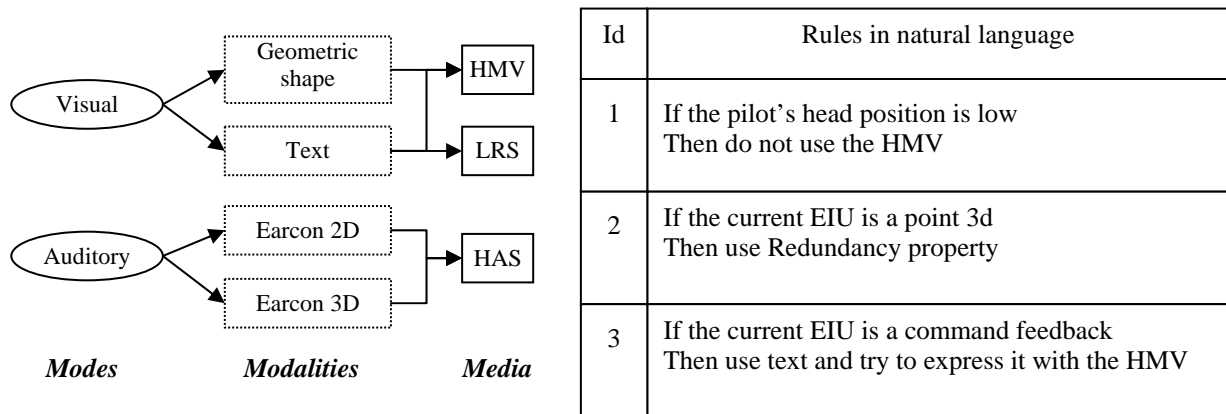
Id	Rules in natural language
1	If the pilot's head position is low Then do not use the HMV
2	If the current EIU is a point 3d Then use Redundancy property
3	If the current EIU is a command feedback Then use text and try to express it with the HMV

Figure 4: Interaction Components Diagram.   Table 1: Rules samples.

The different interaction components managed by the system are described through the Figure 4. 9 interaction components including 2 modes (visual and auditory), 4 modalities (geometric shape, text, earcon 2d and 3d) and 3 media (HMV: Helmet Mounted Visor, LRS: Large Reconfigurable Screen and HAS: Helmet Audio System) are taken into account.

At last, 4 information units (add a valid or invalid mark, refresh and remove a mark) are managed by the system. For example, the information unit "add a valid mark" breaks down into two elementary information units: the mark (point 3d) and the command feedback ("add" operation).

The rules base resulting from this specification is composed of 12 elements. The "influence analyze" step describes in natural language the rules (Table 1). Rule n°1 (Table 1) is described in MOXML in the subsection "Data Representation Language".

Software Architecture Illustration

We have presented how to specify outputs. We will now underline the use of this specification. For that, we will illustrate the software architecture (section 4).

First of all, we fix an interaction context. The pilot has just carried out a valid ground marking out. He is in a nominal situation; the specified context does not influence Computer-Human Interaction.

In a software architecture view, this scenario results in the sending of a message by the dialogue controller to the Multimodal Presentations Management Module (MPMM). This message is an information unit: "add a valid mark" to express. The MPMM transmits the EIU of this information unit ("point 3d" and "command feedback") to the election module.

For each elementary information unit, an election takes place. In the case of the EIU "command feedback", "Text" modality and "HMV" medium pair is elected (rule n°3 Table 1). For the EIU "point 3d", the election behavior is different. Rule n°2 (Table 1) requests an enrichment of the multimodal presentation on the Redundancy criterion. In other words, the multimodal presentation expressing this EIU will be composed of several redundant output modality-medium pairs. In our case, the elected pairs are: Geometric shape – HMV, Geometric shape – LRS and Earcon 3D – HAS. Once the elections carried out, the two allocated multimodal presentations are merged and sent to the MPMM.

At first, the MPMM saves this multimodal presentation like an active presentation. More precisely it is added to the Multimodal Presentations List (MPL). Then, the MPMM transmits a criteria list of the interaction context (used media availabilities, pilot's head position, etc.) to the Spy module. This list describes the interaction context elements which guarantee the presentation validity and need to be supervised. At last, it sends the elected modalities to the different media implied in this presentation. Media instantiate modalities and present to the pilot.

The dialogue controller will then send the information necessary to refresh the current mark. The information unit "refresh mark" does not require an election. This is why the MPMM will only redirect it (without processing) to media included in the mark presentation.

Now, the pilot lowers his head to look at something in the cockpit. The Spy underlines the modification of the interaction context to the MPMM. The MPMM orders to stop any activity to HMV in order to improve LRS visibility. The MPMM calls into question the mark presentation validity and request a new election. The multimodal presentations for the EIU "point 3d" and "command feedback" become respectively: Geometric shape – LRS, Earcon 3D – HAS and Text - LRS. Rule 1 (Table 1) prevented the HMV use. The system adapted to the interaction context by choosing the LRS medium like first display area.

Discussion

There is a major architectural difference between the first and the second application. In the phone application, the adapted multimodal presentation is chosen without worrying about its becoming. The presentation of an incoming call is not indefinitely persistent and only the election engine is really involved. In the case of a plane cockpit, the presentations can have a more significant lifespan and undergo many modifications during their life cycle. This requires the management of the elected multimodal presentations and the use of all architecture modules. In spite of our will to propose a generic architecture model, there is a small dependence to application field which results in a restricted or complete use of our architecture.

These two applications were also created with different objectives. The first application was developed to present our research. Our main motivation was to underline interests to adapt multimodal outputs to the interaction context and to describe in a declarative way the application behavior. The second application is a prototype of the INTUITION project. Objectives were more targeted on the integration within a global architecture and the connection between the various elements in the platform (dialog controller, input controller, etc.).

Concerning the phone application, we have notice that people were above all focussed on contesting the chosen adaptation rules (like "I don't want my phone automatically switches in silent mode in a cinema because…") whereas our objective was not to define a particular behavior but an architecture allowing to specify different application behaviors quickly.

The second application showed some problems related to messages overflow in the dialogue controller. Multi-threading connection between modules enabled us to improve the communication. This application being carried on Thales-Avionics simulator, only few people could be present at the preliminary tests. Nevertheless our industrial partner asked us for graphical editors in order to simplify the outputs specification. These editors are currently under development. A Thales member had some doubts about the contribution of an adaptive system to a military application because of the risks of disturbing and slowing down the pilot. Concerning this point, we think an adaptive system must be used in targeted situations and taught during pilot trainings. Thales-Avionics share this point of view and target the adaptativity use (the prototype behavior is composed of twelve rules).

On the evaluation sight, we break down this step into two parts. The first evaluation relates to the developed applications (mobile phone and ground marking out in a fighter plane). This evaluation was not carried out and will not be carried out by our care. We propose a tool for output multimodal systems design and our participation in the design loop is limited to some council regarding the tool use. We think that only domain application experts are able to evaluate the rules which determine the application behavior. Within the framework of INTUITION project, Thales-Avionics will make this evaluation with experienced pilots at the beginning of the year 2005. The second evaluation relates to the evaluation of the software tool itself and will be done by us at the end of the year 2004.

CONCLUSION

This study allowed us to highlight some needs for the future Human-Computer communications systems in regard to output multimodality. We have proposed a generic architecture model for output multimodal systems. The two implementations draw a beginning of validation of the MOST tool, based on this architecture.

The proposed modules of the software architecture are tested and validated individually. Once tool finalized, it will be necessary to integrate MOST in all tasks of the two simulators of the INTUITION project, which are FACET (a flight simulator of the RAFALE fighter) and MIDDLES (an air traffic control system). This integration will allow to test and validate the tool as a whole.

Lastly, in a longer-term vision, the architecture model proposed will have to be extended and refined. The first extension relates to the election module. It consists in refining the election process in such way to manage the attributes of the different interaction components (mode, modality and medium). Indeed, the attributes of these components can influence the election itself. The second extension consists in extending the properties of the MOST tool to allow instantiation of the elected multimodal presentations. In its current state, MOST is able to allocate a multimodal presentation but it does not carry out the instantiation. Instantiation mechanisms remain to be defined.

REFERENCES

André, E., Finkler, W., Graf, W., Rist, T., Schauder, A. and Wahlster, W. (1993) WIP: The Automatic Synthesis of Multimodal Presentations, Mark T. Maybury editor, *Intelligent Multimedia Interfaces*, 75-93.

Arens, Y., Miller, L., Shapiro, S.C. and. Sondheimer, N.K. (1988) Automatic Construction of User-Interface Displays, *In Proceedings of the 7th AAAI Conference*, St. Paul.

Arens Y. and Hovy E. (1995) The Design of a Model-Based Multimedia Interaction Manager, *AI Review*, 9, 3.

Bellik Y. and Farcy R. (2002) Comparison of Various Interface Modalities for a Locomotion Assistance Device, *8th International Conference on Computers Helping People with Special Needs*, Austria, July.

Bernsen N.O. and Dybkjaer L. (2001) Exploring Natural Interaction in the Car, *Workshop Information Presentation and Natural Multimodal Dialogue*, Verona, Italy, 75-79.

Braffort, A., Choisier, A., Collet, C., Dalle, P., Gianni, F., Lenseigne, B., Segouat, J. (2004) Toward an annotation software for video of Sign Language, including image processing tools and signing space modelling, *4th International Conference on Language Resources and Evaluation*, Lisbonne, Portugal.

Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J. and Young, R. (1995) Four easy pieces for assessing the usability of multimodal interaction: The CARE properties, *In Proceedings of Interact'95*, 115-120.

Dalal, M., Feiner, S., McKeown, K., Pan, S., Zhou, M., Höllerer, T., Shaw, J., Feng, Y. and Fromer, J. (1996) Negotiation for Automated Generation of Temporal Multimedia Presentations. *ACM Multimedia 96*, 55-64.

Dey, A.K. (2001) Understanding and using context. Personal and Ubiquitous Computing, *Special issue on Situated Interaction and Ubiquitous Computing 5*, 1.

Feiner S and McKeown K. (1993) Automating the Generation of Coordinated Multimedia Explanations, Mark T. Maybury editor, *Intelligent Multimedia Interfaces*, AAAI Press / MIT Press, 117-139.

Grynszpan, O., Martin, J.-C. and Oudin, N. Towards (2003) A Methodology For The Design Of Human-Computer Interfaces For Persons With Autism, *International Congress Autism-Europe*, Lisboa.

Gu, T., Wang, X.H., Pung, H.K. and Zhang, D.Q. (2004) An Ontology-based Context Model in Intelligent Environments, *In Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, San Diego, USA.

Jackson, P. (1999) *Introduction to Expert Systems*, Harlow: Addison-Wesley Longman Limited.

Neal J. G. and Shapiro S.C. (1991) Intelligent Multi-Media Interface Technology, *Intelligent User Interfaces*, ACM press, New York, USA, 11-43.

Rist T. and Brandmeier P. (2002) Customising Graphics for Tiny Displays of Mobile Devices, *Personal and Ubiquitous Computing*, 6, 260-268.

Stephanidis C. and Savidis A. (2001) Universal Access in the Information Society: Methods, Tools, and Interaction Technologies, *UAIS 2001*.

Thevenin, D. (2001) Adaptation en Interaction Homme-Machine : le cas de la Plasticité, Ph.D. Thesis, University Joseph Fourier.

Rousseau, C. (2003) Etude d'un modele pour une interaction Homme-Machine dynamique et contextuelle, DEA report, University Paris XI.

ACKNOWLEDGEMENTS

COPYRIGHT