TYCOON: Theoretical Framework and Software Tools for Multimodal Interfaces

Jean-Claude Martin

We define a *modality* as a process analyzing and producing chunks of information. For instance, a speech recognition modality analyses speech signals and produces the labels of recognized words. Several multimodal interfaces combining such modalities have already been developed (IMMI'95, CMC'95). To take benefit out of them so as to advance research and implementation of multimodal interfaces, coherent theoretical and software tools are needed.

From the "theoretical" point of view, the development of multimodal interfaces addresses several issues (Maybury 91, Dowell 95): content selection ("what to say"), modality allocation ("which modality to say it"), modality realization ("how to say it in that modality") and modality combination. This paper deals with the "modality combination" issue. A multimodal interface developer has to know how to combine modalities and why this combination may improve the interaction. Yet existing frameworks for human-computer interfaces do not answer these two questions. Instead, they deal with the relation between the modes (language or action), the channels (audio, visual or haptic), the media (speech, text or gesture) and the styles of interaction (command language, selection in a menu) (Frohlich 91). Other frameworks describe the specificities of each modality regarding information content (Bernsen 95) or the temporal and semantic relations between events detected on several modalities (Nigay & Coutaz 93, Karagiannidis 95).

From the "software tools" points of view, existing authoring tools enable only the multimedia developer to combine modalities on temporal and spatial dimensions. A common deficiency of these tools is the lack of support mechanisms for the design and implementation tasks (Väänänen 95).

This paper describes our approach named TYCOON, which is based on the notion of TYpes and goals of COOperatioN between modalities. It covers both the theoretical and the software points of view. It is composed of a theoretical framework for studying multimodal interfaces, a specification language and a multimodal module integrating events detected by several modalities.

A Theoretical Framework for Studying Multimodal Interfaces

A study of the literature on multimodality in Psychology, Neurobiology, Artificial Intelligence and Human-Computer Interaction (Martin 95) led us to distinguish five basic types of cooperation between modalities, respectively named: *transfer*, *equivalence*, *specialization*, *redundancy* and *complementarity*. These types of cooperation can be viewed as different rules for combining modalities. In this section we define them and we describe how each of them may be brought into play for several goals which constitute a second dimension of the framework (figure 1).

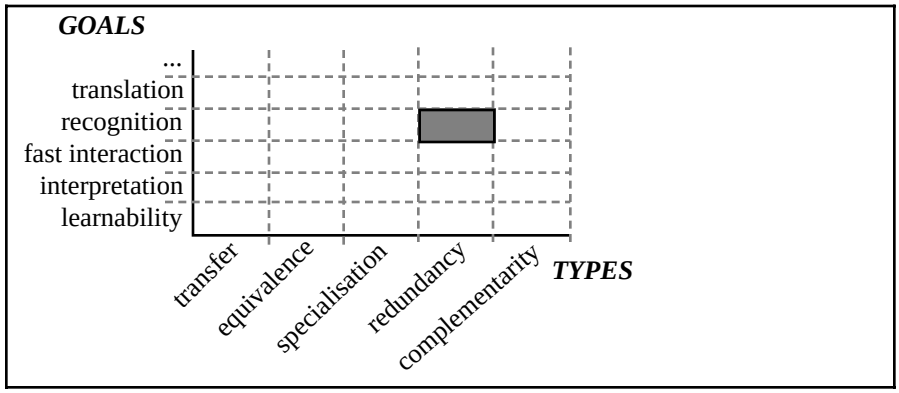

Figure 1. A theoretical framework for studying multimodality. Modalities may cooperate with one another according to several types of cooperation (x-axis). Each of these type may be involved in several goals for improving human-computer interaction (y-axis). As an example, the dashed square represent the fact that redundancy between two modalities may be used to improve recognition.

Transfer

When several modalities cooperate by *transfer*, this means that a chunk of information produced by a modality is used by another modality. Transfer is commonly used in hypermedia interfaces when a mouse click provokes the display of an image (Inder et al. 95). In information retrieval applications, the user may express a request in one modality (speech) and get relevant information in another modality (video) (Foote et al. 95). Output information may not only be retrieved but also produced from scratch: several systems generate a graphical description

of a scene from a linguistic description (O Nuallàin and Smith 94, Olivier and Tsujii 94). Similarly, the visual description of a scene can be used to generate a linguistic description (Jackendoff 87, Daniel et al. 94) or a multimodal description (André and Rist 95). Let's say that all these previous examples involved transfer for a goal of *translation*.

Transfer may also be involved in other goals such as *improving recognition*: mouse click detection may be transferred to a speech modality in order to ease the recognition of predictable words (here, that...) as in the GERBAL system (Salisbury et al. 90).

Transfer may also be used to *enable a faster interaction*: in the MAAR system (Cheyer and Julia 95), when part of an uttered sentence has been misrecognized, it can be edited with the keyboard so that the user does not have to type or utter again the all sentence. Finally, the WIP system (Wahlster et al. 91) produces coordinated natural language and graphics output. The two modalities work concurrently to produce an output based on instructions received from a multimodal manager. If necessary, when one of the modalities cannot produce a given piece of information, on-line information can be transferred from this modality to the other. As an example, the graphical modality can be told by the manager to produce a textual label and it may turn out that it is not possible because it would hide parts of the graphics. This information is sent to the natural language modality which is able to adapt dynamically its output to insert the new textual information. This transfer of information enables parallelism of processing in two modalities and hence a faster human-computer interaction.

Thus, transfer may intervene for different reasons either between two input modalities, or between two output modalities, or between an input modality and an output modality.

Equivalence

When several modalities cooperate by *equivalence*, this means that a chunk of information may be processed as an alternative, by either of them.

For instance, in the TAPAGE system (Faure and Julia 94), the user of a graphical editor may specify a command either through speech or through the selection of a button with a pen. In this case, equivalence enables the user to select a command with the pen when the speech recognizer is not working accurately because of noise, and hence to *improve recognition* of the commands.

Equivalence also enables *adaptation to the user* by customization: the user may be allowed to select the modalities he prefers (Hare et al. 95). The formation of accurate mental models of a multimodal system seems dependent upon the implementation of such options over which the user has control (Sims and Hedberg 95). Equivalence also enables a *faster interaction* since it allows the system or the

user to select the fastest modality. Thus, equivalence means alternative. It is clear that differences between each modality, either cognitive or technical, have to be considered.

Specialization

When modalities cooperate by *specialization*, this means that a specific kind of information is always processed by the same modality. In fact, specialization is not always so absolute and may be more precisely defined: one should distinguish *data-relative* specialization and *modality-relative* specialization. For example, in several existing systems, sounds are somehow specialized in errors notification (forbidden commands are signaled with a beep). It is a modality-relative specialization if sounds are not used to convey any other type of knowledge. It is a data-relative specialization if errors only produce sounds and no graphics or text. When there is a one-to-one relation between a set of information and a modality managing this set, we will speak of *absolute specialization.*

This specialization may help the user to *interpret* the events produced by the computer (to link them to the global contextual knowledge). This means that the choice of a given modality adds semantic information and hence helps the interpretation process.

Specialization may also *improve recognition*. In the example of a tourist information system, the user may always provide the name of towns using the keyboard. This specialization enables an easier processing (and hence a better recognition) in other modalities. It improves the accuracy of the speech recognizer since the search space is smaller (Baekgaard 95).
This may also enable a faster interaction since it decreases the duration of the integration and modality selection process.

When a modality is specialized, it should respect the specificity of this modality including the information it is good at representing. For instance, in reference interpretation, the designation gesture aims at selecting a specific area and the verbal channel provides a frame for the interpretation of the reference: categorical information, constraints on the number of objects selected (Bellalem and Romary 95).

Intuitive specialization of a modality may goes against its technical specificities. In the Wizard of Oz experiment dealing with a touristic application described in (Siroux et al. 95), despite the low recognition rate of town names, the users did not use the tactile screen to select a town but used speech instead.

Redundancy

If several modalities cooperate by *redundancy*, this means that the same information is processed by these modalities. For instance, if the user types “quit” on the

keyboard and utters “quit," this redundancy can be used by the system to avoid a confirmation dialogue and thus enables a *faster interaction*.

Regarding *intuitiveness*, redundancy has been observed in the Wizard of Oz study described in (Siroux et al. 95): sometimes the user selected a town both by speech and a touch on the tactile screen.

Regarding *learnability* of interfaces, it has been observed that a redundant multimodal output involving both visual display of a text and speech restitution of the same text enabled faster graphical interface learning (Wang et al. 93). An important issue here is to know if the visual channel should carry exactly the same message as the auditory channel (verbatim reinforcement) or a shorter one (priming reinforcement). The type of reinforcement chosen by the system and the information to be transmitted seem to have consequences of the cognitive compatibility of spoken or manual responses from the user (Dowell et al. 95). Redundancy between visual and vocal text with verbatim reinforcement was also tested in (Huls and Bos 95) with natural language descriptions of the objects the user manipulates and the action he performs. Although speech coerced the subjects into reading the typed descriptions, the subjects made more errors and were slower than with the visual text output only.

Complementarity

Finally, when several modalities cooperate by *complementarity*, it means that different chunks of information are processed by each modality but have to be merged. First systems enabling the “put that there” command for the manipulation of graphical objects are described in (Carbonnel 70, Bolt 80).

This complementarity enables a *faster interaction* since the two modalities can be used simultaneously and convey shorter messages which are also *better recognized* than longer messages.

Complementarity may also improve *interpretation*, as in (Santana and Pineda 95) where a graphical output is sufficient for an expert but need to be completed by a textual output for novice users.

An important issue concerning complementarity is the criterion used to merged chunks of information in different modalities. The most classical approaches are to merge them because they are temporally coincident, temporally sequential or spatially linked. Regarding intuitiveness, complementarity behavior were observed in (Siroux et al. 95). Two types of behavior did feature complementarity: the sequential and the synergetic behaviors. In the “sequential” behavior, which is rare, the user would by example utter “what are the campsites at” and then select a town with the tactile screen. In the “synergetic” behavior, the user would utter “Are there any campsites here ?” and select a town with the tactile screen while pronouncing “here." Regarding the output from the computer, it was observed in

the experiment described in (Hare et al. 95) that spatial linking of related information encourages the user's awareness of causal and cognitive links. Yet, when having to retrieve complementary chunks of information from different media, users behavior tended to be biased towards sequential search avoiding synergetic use of several modalities.

Modalities cooperating by complementarity may be specialized in different types of information. In the example of a graphical editor, the name of an object may be always specified with speech while its position is specified with the mouse. But modalities cooperating by complementarity may be also be equivalent for different types of information. For instance, the user may select an object with the mouse and its position with speech ("in the upper right corner"). Nevertheless, the complementary use of specialized modalities gives the advantages of specialization: speech recognition is improved since the vocabulary and syntax is simpler than a complete linguistic description such as "put the red square which is on the left hand side above the green rectangle." As an example, the use of the natural complementarity of the speech audio and the images of the lips movements improves speech recognition (Vo and Waibel 93).

Discussion

These types of cooperation (excluding transfer) can be compared through the two dimensions *fusion* and *transmitted information* (table 1):

- equivalence and specialization exclude fusion,
- redundancy and complementarity require fusion,
- equivalence and redundancy require transmission of the same information,
- specialization and complementarity require transmission of different information.

	Same information	Different information
No fusion	Equivalence	Specialization
Fusion	Redundancy	Complementarity

Table 1: comparison of equivalence, specialization, redundancy and complementarity.

In fact, the chunks of information considered in these types of cooperation can be of different levels of abstraction depending on the modality. For instance, the semantic interpretation of an uttered sentence “clear all the screen” may be equivalent to a lexical entry with the keyboard (a function key).
The dimension “types of cooperation” initially introduced in (Martin and Béroule 93) has been used and renamed “the CARE properties” in (Nigay and Coutaz 95). However, some features of TYCOON are missing such as the type “transfer," the distinction between data-relative and modality-relative specializations and other goals such as “enabling a fast interaction."

In this section we have described our framework, defined its terminology and showed how it can be used to study existing multimodal interfaces. Formal notations of the types of cooperation can be found in (Martin 95). In the next section we describe examples taken from COMIT, a multimodal interface that we have developed.

A Multimodal Interface for the Building of Graphical Interfaces

COMIT features several types and goals of cooperation between a Datavox-Vecsys speech recognition system, a keyboard and a mouse. The events detected by the three modalities are time-stamped in a coherent fashion by a Modality Server (Krus 95, Bourdot et al. 95) and integrated by a multimodal module. This multimodal module interprets sequences of detected events as commands which are executed by the MOTIF application editing graphical interfaces (figure 2). In this section, we give examples of multimodal commands and their specification. Details on the corresponding processing in the multimodal module will be explained in the next section.

Several Possible Fusion Criteria

In COMIT, the user can create a graphical button called “OK” at the graphical location specified by the mouse by producing the sequence of events of figure 3. We hereafter explain the specification of this multimodal command involving three variables V1, V2 and V3:

specialization V1 SPEECH button

Creates a first variable V1 which will be activated if the word “button” is recognized by the speech modality.

complementarity_coinc V2 SPEECH called KEYBOARD *

Creates a variable V2 involving a cooperation by complementarity with a temporal coincidence criterion which enables the word “called” on the speech modality to be merged with any word typed on the keyboard in the same temporal window.

*complementarity_coinc V3 SPEECH here MOUSE click **

Creates a variable V3 which also involves a cooperation by complementarity and enables the word “here” to be merged with a mouse click at any location in the same temporal window.

complementarity_sequence V1 V2 V3

The three variables are linked sequentially.

coincidence_duration 1500

The length of the temporal window is specified ; the given value (1500) is interpreted as a number of processing cycles of the multimodal module.

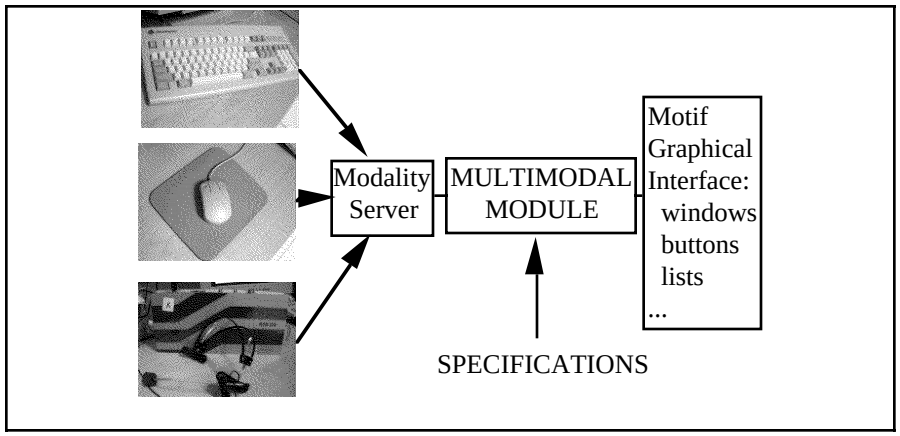

Figure 2. Architecture of the multimodal interface COMIT. Events detected by the keyboard, the mouse and the speech recognizer are time-stamped by a modality server (Krus 95). The events are then integrated by a multimodal module which activates command of the application (editor of graphical interface). The cooperations between modalities are specified with a command language which is used to define COMIT.

As another example, the user can create a list of selectable words. Since the number of words is a priori unknown of the system, it is not possible to use a temporal coincidence criterion to merge them. Instead, the user can specify the end of the list by uttering “end of list” (Figure 4). The specification of this command involves what we call a “structure completion” criterion enabling the fusion of all the words typed before the utterance of “end of list”:

```
special V4 SPEECH list
complementarity_structural V5 KEYBOARD *
SPEECH end_of_list
complementarity_sequence V4 V5
```

Fast Interaction

Simultaneous or overlapping independent commands can be recognized in parallel in COMIT. For instance, the user may start creating a button then ask for the date and finally finish the button command (figure 5).

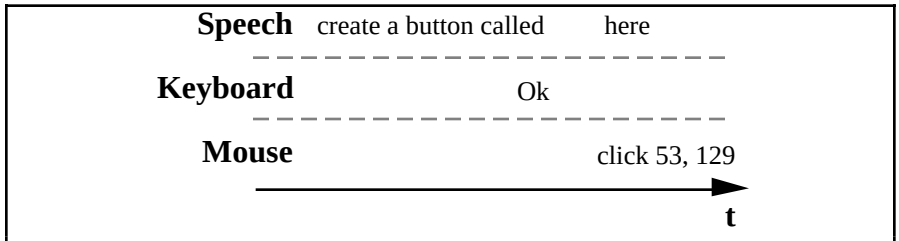

Figure 3: Events produced by the user to create a button.

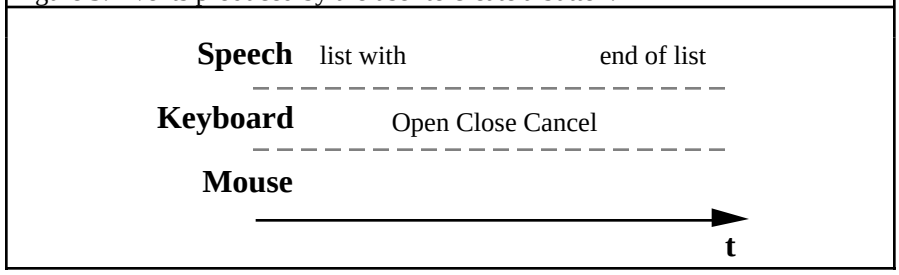

Figure 4: Events produced by the user to create a list.

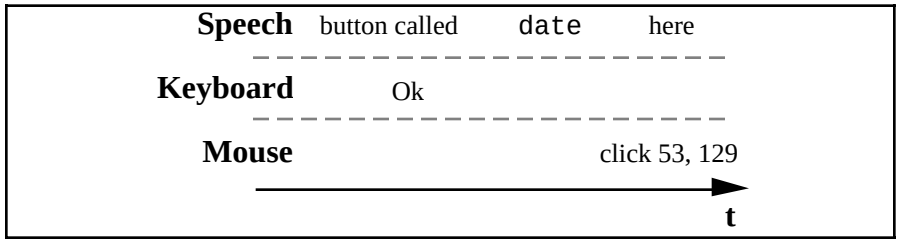

Figure 5: COMIT is able to interpret two overlapped independent commands such as the creation of a button and a command asking the date.

Redundancy also enables a faster interaction in COMIT. When the command “quit” is detected only on speech or keyboard, COMIT asks for a confirmation. When it is detected on both speech and keyboard modalities within the same temporal window, COMIT does not ask for a confirmation and directly quits the application (figure 6).

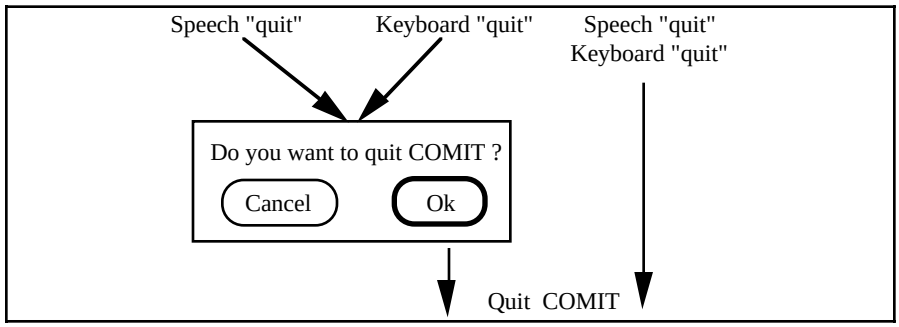

Figure 6: COMIT uses redundancy to avoid a confirmation dialogue and thus enables a faster interaction.

Improving Recognition

In COMIT, transfer is used to improve recognition through predictions. The activation of a variable leads to a transfer of information (decrease threshold of pre-

dicted events). Table 2 describes some predictions in the example of the creation of a button. The word “button” is detected (first line). One of the predictions made by the system is that the word “called” is going to be detected (second line). This prediction provokes a decreasing of its recognition threshold. The detection of the word “OK” on the keyboard enables another prediction which decreases the threshold of “called” down to 0.7. When the word “called” is detected with a low probability of 0.73 (bottom line), it is considered as recognized and propagates information in the network. This would not have been possible without predictions.

EVENT Speech “button”
PREDICTION of “called” Threshold 0.9 -> 0.8
EVENT Keyboard “OK”
PREDICTION of “called” Threshold 0.8 -> 0.7
EVENT Speech “called," Probability 0.73

Table 2: Improving recognition thanks to multimodal predictions.

The recognition done in COMIT is not only robust to noisy events but also somehow to missing, inverted or repeated events. When the sequence of events corresponds exactly to the specification, the recognition score is at a maximal value (1.0). When events appear a bit sooner or later than expected, the recognition score of the command is lower (0.83). If some events are missing, the score is also lower than 1.0. For instance in the command creating a button (figure 3), if the word “called” is not detected by the speech recognition system, this command is nevertheless recognized but with a lower score of 0.721. If both the word “called” and “here” are not detected, the score is 0.665. If the mouse click is also missing, the score is 0.589. The bigger the difference between the specifications and the sequence of detected events, the lower the recognition score. This continuous recognition score enables the recognition of the command without having the user to produce again the same sequence of events. It also provides the system with a criterion for solving ambiguities. When several commands have a non-zero recognition score with the same sequence of events, the command which is executed is the one with the greatest score.

In the next section, we explain how the multimodal module enables these properties of COMIT.

A Multimodal Module

From the specifications, the multimodal module builds a *Guided Propagation Network* (Béroule 85) which brings into play elementary interconnected processing units exchanging signals. Current monomodal applications of these networks include: speech recognition (Escande et al. 91), strategies of syntactic learning (Roques 94), robust parsing (Westerlund et al. 94), hand-written character recognition and modeling of reading (Béroule et al. 94). In COMIT, this network contains one *event-detector* for each expected event on each modality. When an event is detected, the associated event-detector is activated and sends signals to one or several *variable* units.

Several Possible Types of Cooperation

These networks enable the management of several types of cooperation between modalities. In the actual system, only cooperations at a lexical level are possible. For each command of the specification language, we describe the corresponding network and its dynamics during recognition.

Equivalence: creates a variable unit which may be activated by one event-detector or another (figure 7).

Specialization: creates a variable-unit which can be activated by only one event-detector (figure 8).

Redundancy: creates a variable-unit which can be activated by two event-detectors having the same name in different modalities. The only possible fusion criterion for redundancy in the actual system is temporal coincidence (figure 9).

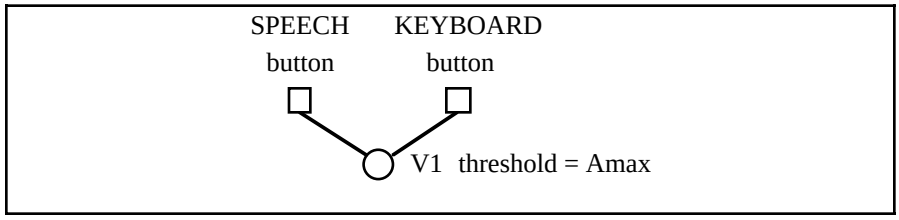

Figure 7.a. Network created by the command equivalence V1 SPEECH button KEYBOARD button. A variable unit (circle) is created and connected to two event detectors (squares).

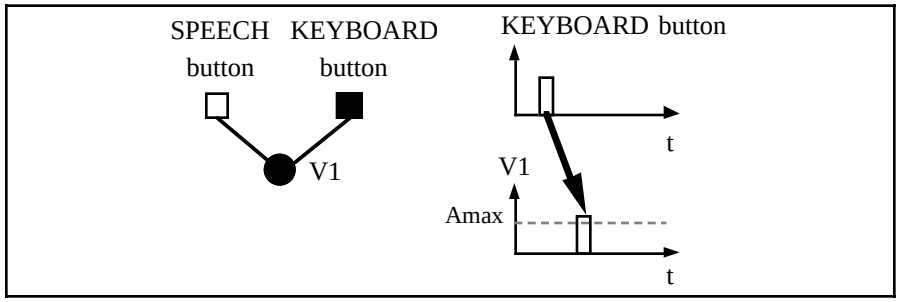

Figure 7.b. Dynamical processing of equivalence. The word "button" has been typed on the keyboard and the corresponding detector is activated (left-hand side). The histograms on the right-hand side show the variations of activity of this detector and of V1 as a function of time. The activated detector sends a signal of amplitude Amax to V1 (dark arrow). V1 becomes activated above its threshold (dotted line).

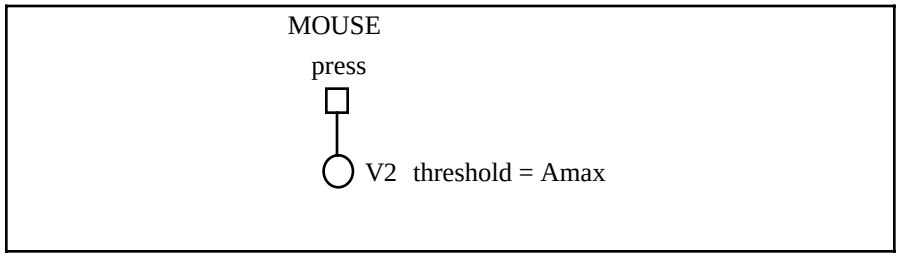

Figure 8. Network created by the command specialization V2 MOUSE pressV2 can be activated by the Mouse press detector as in figure 7.b.

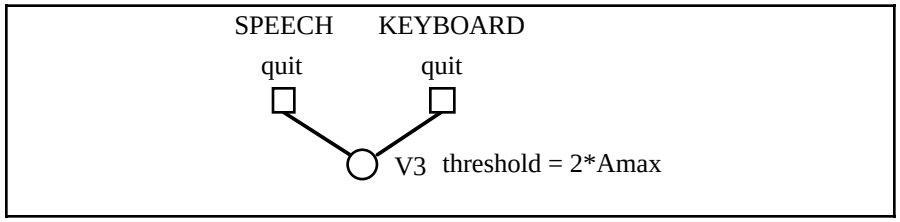

Figure 9.a. Network created by the command redundancy V3 quit SPEECH KEYBOARD

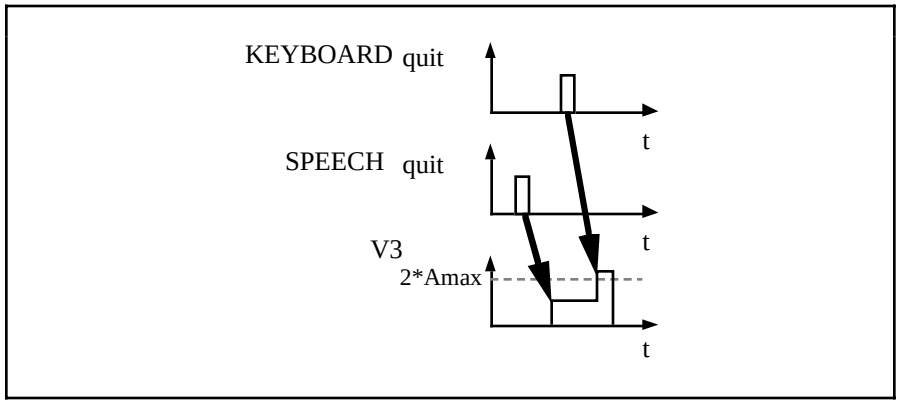

Figure 9.b. Dynamical processing of redundancy and temporal coincidence. The word "quit" has been uttered and then typed on the keyboard within the same temporal window. The "quit" detector of the speech modality first sends a signal to V3 which becomes activated below its threshold. This signal will only last for a finite duration which is the width of the temporal window. When a signal is sent by the detector "quit" of the keyboard within this temporal window, this signal is added to the signal emitted by the speech modality and thus enable V3 to be activated above its threshold.

Complementarity: creates a variable-unit which can be activated by several event-detectors. The possible fusion criteria in the actual system are temporal coincidence, temporal sequence and structural completion (figure 10).

Transfer: enables the transfer of signals between units (figure 11).

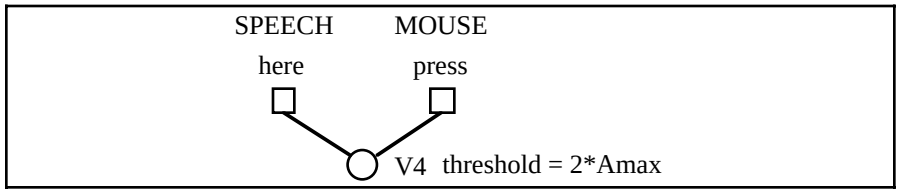

Figure 10.a. Network created by the command complementarity_coinc V4 MOUSE press SPEECH here

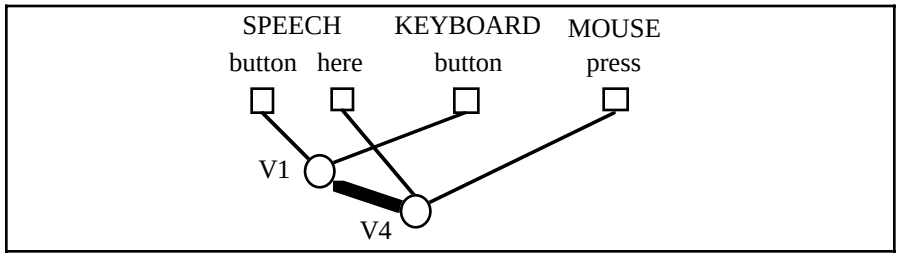

Figure 10.b. Network created by the command complementarity_sequence V1 V4 A temporal link (thick line) is created between already existing V1 and V4.

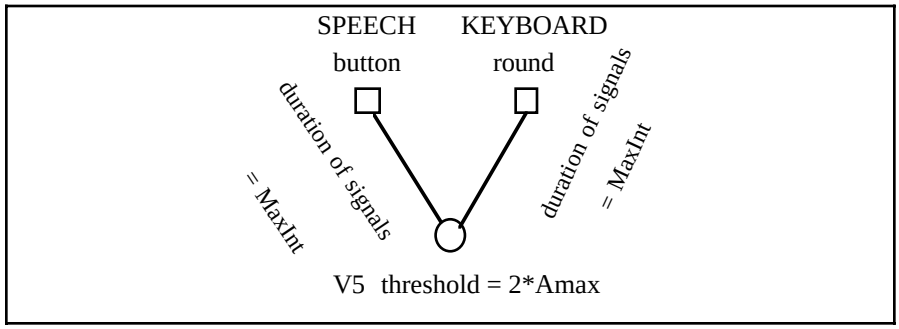

Figure 10.c. Network created by the command complementarity_structural V5 SPEECH button KEYBOARD round. It creates a variable-unit V5 which is activated

when the word button is pronounced and the word round is typed, whatever is the time interval between these two events. This is enabled by setting the duration of the signals emitted by the event-detectors linked to V5 to an infinite value.

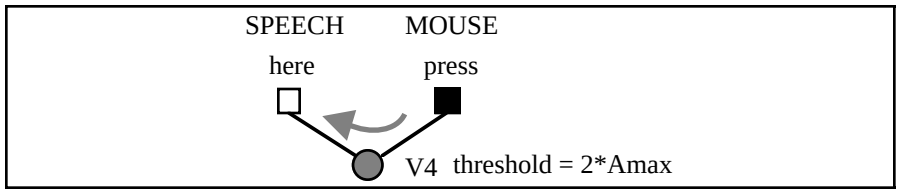

Figure 11. The commands transfer MOUSE SPEECH and goal ImproveRecognition enable the transfer of information from the mouse modality to the speech modality to improve speech recognition. V4 has been specified as in figure 10. When a mouse click is detected, the recognition threshold of the word here on the speech modality is lowered.

Fusion and Interpretation of Variable Events

The previous examples were dealing with constant events. Anyway, a multimodal command often feature variable events which change from one instance of this command to another (for instance, the name or the position of a button to create). To deal with such events, mechanisms for coding fusion results have been added to Guided Propagation Networks (figure 12).

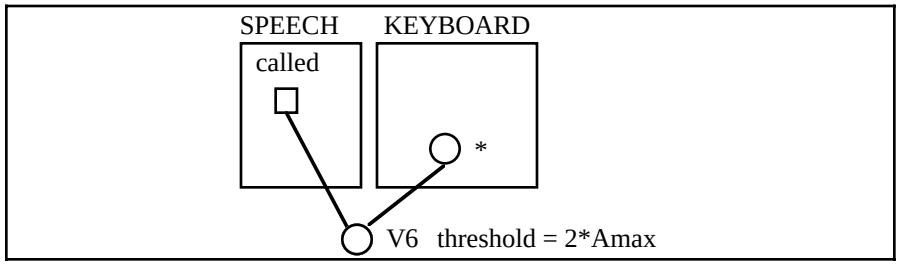

*Figure 12.a. Network created by the specification complementarity_coinc V6 SPEECH called KEYBOARD **

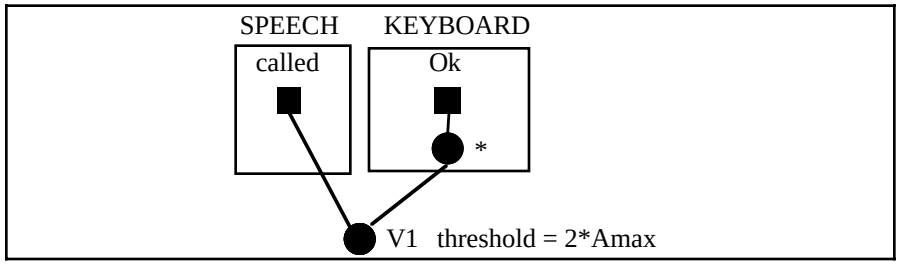

*Figure 12.b. During the execution, when a variable event is detected (a non keyword typed on the keyboard as "OK"), a corresponding event-detector is created and linked to the * unit. The unit V1 gets activated if the unit "called" and the unit "*" are activated in the same temporal window*

Figure 13 details these mechanisms during the recognition of the whole command creating a button. In order to execute the recognized command, the representations of the network have to be bound to a representation of the application. The command "bind_application" is managing this operation. As an example, the command for creating a button is managed by the following procedure of the application:

CreateButton (Param1CreateButton , Param2CreateButton)

The link between this procedure and the variables is specified by:

end_command V3 CreateButton

bind_application Param1CreateButton V2

bind_application Param2CreateButton V3

The "bind_application" commands bind the variable units V2 (the name of the button) and V3 (the position where to put the button) to the corresponding parameters of the CreateButton command. During execution, the binding between the actual values of these variables and the two parameters are done with the same temporal coding principles used for coding fusion results in figure 13.

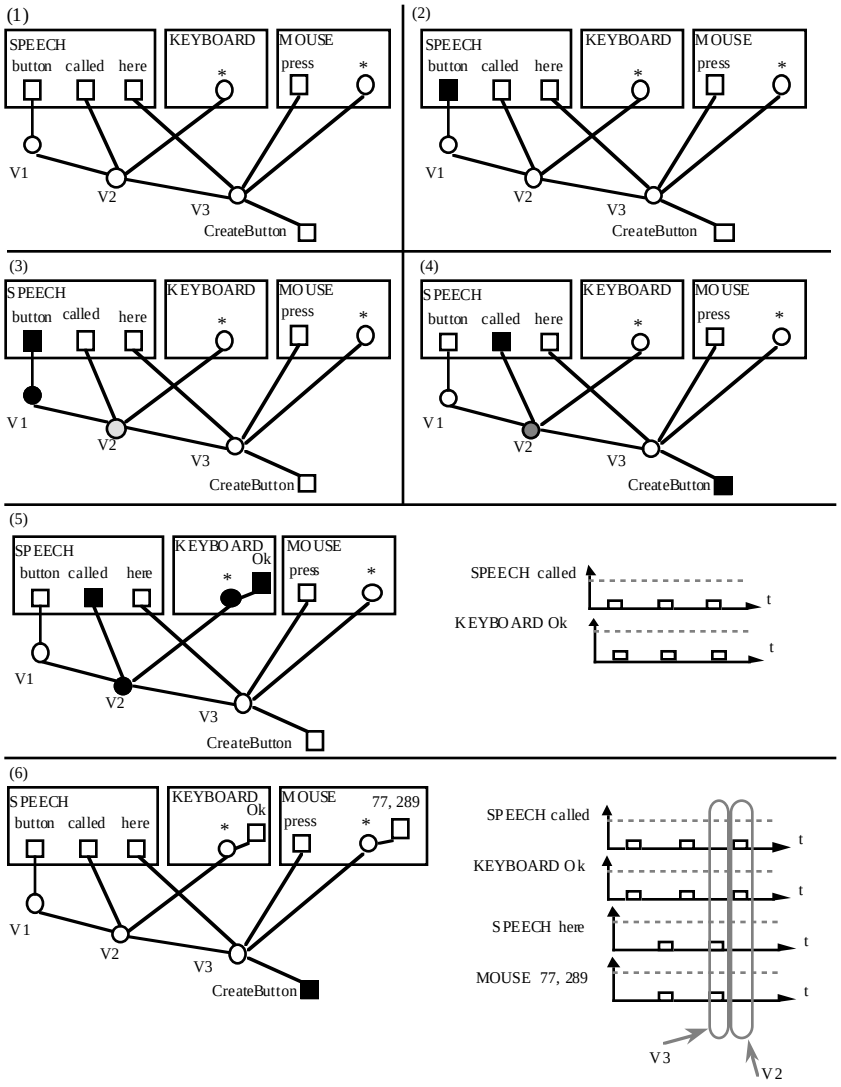

Figure 13. Representation and recognition of a multimodal command in COMIT.

The command creating a button in Figure 13 is represented by a pathway linking event detectors (squares) and three variable units (circles): V1, V2 and V3.

(1) Initially, no unit is activated

(2) The word “button” is recognized and its associated detector is fully activated. This detector sends a signal to variable V1

(3) V1 becomes activated and sends a signal to variable V2 which becomes activated below its threshold (in grey)

(4) The word “called” is detected and the corresponding detector is activated. V2 becomes more activated (dark grey)

(5) The work “OK” has been typed on the keyboard. A corresponding detector has been created and linked to unit “*." V2 becomes fully activated. This gates a dynamic binding process which results in the emission of synchronized pulses by the event detectors which participated in the activation of V2 (histograms on the right-hand side).

(6) Since a distinct phase is associated to distinct variables, the bindings are readable without cross-talks.

Several Possible Goals of Cooperation

In this section, we explain how the features of these networks enable COMIT to cover two goals of cooperation: “fast interaction” and “improving recognition."

As we stated in the previous section, COMIT is able to deal with independent overlapped commands. This property is due to the parallel calculation of the activities in the network (figure 14).

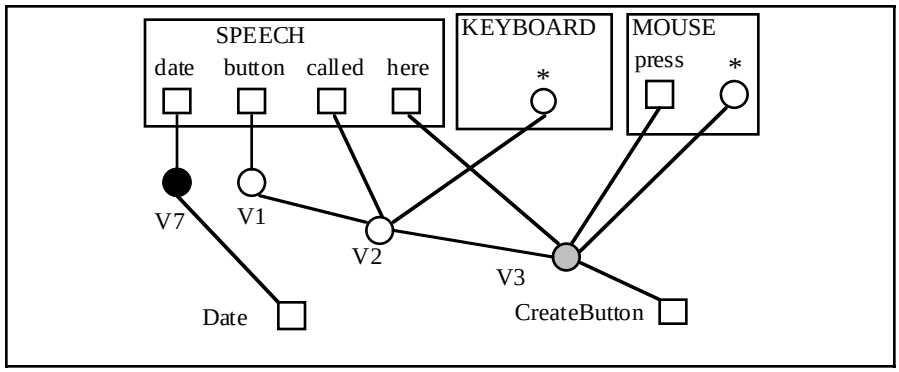

Figure 14. Independent overlapped commands of figure 5 lead to separate flows of activities: V3 slightly activated, V7 fully activated)

The possibility to use redundancy to avoid confirmation dialogue for quitting the application is specified by:

```
equivalence V8 SPEECH quit KEYBOARD quit
end_command V8 QuitWithConfirmation
redundancy V9 quit SPEECH KEYBOARD
end_command V9 QuitWithoutConfirmation
```

These specifications lead to the network of figure 15.

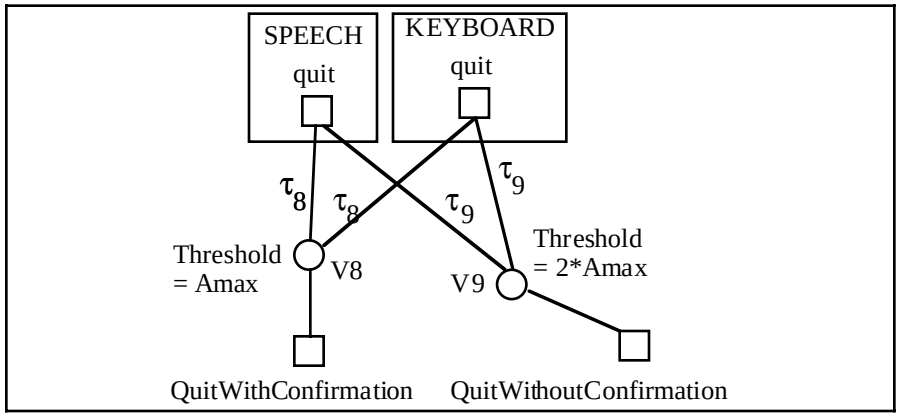

*Figure 15. Redundancy to avoid a confirmation dialogue. Variable V8 can be activated either by speech or the keyboard (threshold = Amax). V9 is activated only if "quit" is detected by both modalities (threshold = 2*Amax). To allow effective detection of coincidence, V9 has a higher priority than V8. This is achieved by delaying the signals activating V8 by τ8 = τ9 + constant. A signal emitted by a "quit" detector, if not used for activating V9 is then used to activate V8.*

To improve recognition, COMIT makes use of predictions. These predictions are realized by the following way: when a variable unite is slightly activated either by another variable unit or an event detector, the threshold of all the event detectors connected to this variable unit is decreased.

The capacity of the multimodal module to provide multimodal recognition scores is due to three features of the networks. The first one is related to the in-

put/output function of each variable unit. The signals received by a variable-unit V_i are: Si, the stimuli emitted by event-detectors, and C_{i-1}, the signals emitted by another variable unit towards V_i. A parameter R_{i-1} is used to tune the respective contributions of S_i and C_{i-1} in the input of V_i: $S_i+R_{i-1}*C_{.i-1}$. To get the recognition scores described in the previous section, each unit makes use of a linear transfer function (figure 16) and the Ri parameters have to follow an arithmetic progression from the first unit of a pathway until the command detector at the end of this pathway (Martin et al. 95).

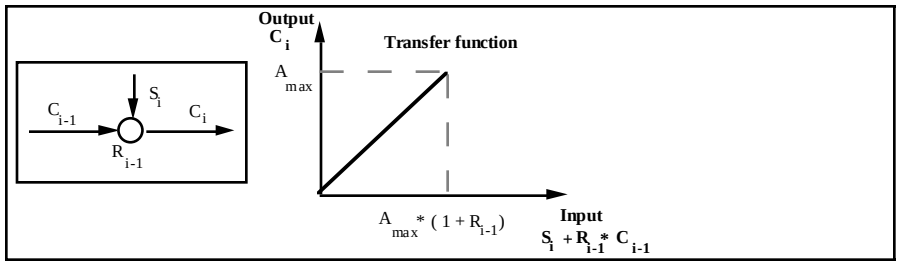

Figure 16. To enable multimodal recognition score, the output of each variable unit is proportional to its input.

The second feature enabling multimodal recognition scores is the “fuzzy” function used in temporal coincidence detection. The signals emitted by the event detectors have a decreasing amplitude. Thus, the closer are two events, the bigger the sum of the signals emitted by their event detectors. Thus, the user does not have to be too strict on his temporal behavior and the width of the temporal window which is difficult to fix, is not so important. Most of the existing multimodal systems use a strict temporal coincidence.

Finally, the amplitude of the signals emitted by the event detectors is proportional to the recognition score of this event. Thus, the multimodal recognition score takes into account the score given by the speech recognizer.

Conclusion

In this paper, we have described out approach named TYCOON for studying and developing multimodal interfaces, trying to get a full benefit of multimodality. It is composed of three parts : a theoretical framework for studying multimodal interfaces, a command language for specifying cooperation between modalities and a multimodal module for integrating events detected on several modalities.

Is this approach applicable to any applications ? Regarding the theoretical framework, the dimension “goals of cooperation” may be easily adapted to other requirement specifications. But since then, we did not found any other possible type of cooperation which should be added in the dimension “types of cooperation." Considering the lacks of existing multimedia authoring tools, the framework could be useful by providing references to experimental results, even if the multimodal developer would have to evaluate how these results can be applied to his own needs. It should be noted that the framework can also be used dynamically: depending on the most urgent current goal (enabling a fast interaction in emergency situations), the set of available types of interaction may evolve. Regarding the specification language and the multimodale module, both are currently being applied to another application: a multimodal editor of conceptual graphs. The capacities of the command language and the multimodal module are going to be extended to include more complex syntactic and semantic processing.

Acknowledgments

Jean-Claude Martin was financed by a Dret-CNRS grant and wishes to thank Dominique Béroule for useful comments on this paper.

References

André, E. and Rist, T. 1995. Generating coherent presentations employing textual and visual material. *Artificial Intelligence Review* 9 (2-3), pp 147-165.

Baekgaard, A. 1995. Constraining of input media in a spoken dialog system. In Proceedings of the fourth european Conference on Speech Communication and technology (EUROSPEECH’95). pp 1181-1184. Madrid, September. ISSN 1018-4074

Bellalem, N. and Romary, L. 1995. Reference interpretation in a multimodal environment combining speech and gesture. In IMMI 95.

Bernsen, N.O. 1995. Information mapping in practice. Rule-base multimodal interface design. Page HTML.

Béroule, D. 1985. An adaptative, dynamic and associative memory model for automatic speech processing. PhD thesis. may 31, Orsay. In French.

Béroule, D., Von Hoe, R., Ruellan, H. (1994). A Guided Propagation Model of Reading, IPO, Annual Progress Report N°28, Eindhoven, Netherlands.

Bolt, R.A. 1980. “Put-That-There”: Voice and Gesture at The Graphics Interface. *Computer Graphics* 14 (3):262-270.

Bourdot, P.; Krus, M.; and Gherbi, R. 1995. Management of non-standard devices for multimodal user interfaces under unix/X11. In Proceedings of the International Conference

on Cooperative Multimodal Communication (CMC/95). part I. pp 49-62. Eindhoven, may 24-26.

Carbonnel, J.R. (1970). Mixed-Initiative Man-Computer Dialogues. Bolt, Beranek and Newman (BBN) Report N° 1971, Cambridge, MA.

Cheyer, A. and Julia, L. 1995. Multimodal maps: an agent-based approach. In Proceedings of the International Conference on Cooperative Multimodal Communication (CMC/95). part I. pp 103-113. Eindhoven, may 24-26. http://www.ai.sri.com/~julia/.

CMC 1995. Proceedings of the International Conference on Cooperative Multimodal Communication (CMC/95). Eindhoven, may 24-26.

Daniel, M.P., Carite, L. and Denis, M. (1994) Modes of linearization in the description of spatial configurations. In Portugali, J. (ed.), The construction of cognitive maps. Kluwer: Dordrecht, The Netherlands.

Dowell, J.; Shmueli, Y.; and Salter, I. 1995. Applying a cognitive model of the user to the design of a multimodal speech interface. In IMMI 1995

Escande, P.; Béroule, D.; and Blanchet, P. 1991. Speech Recognition Experiments with Guided Propagation. Proc. of IJCNN'91, Singapour.

Faure, C. and Julia, L. 1994. An Agent-Based Architecture for a Multimodal Interface. InWorking notes of the AAAI Intelligent Multi-Media Multi-Modal Systems Symp. Standford Univ., 21-23 march.

Foote, J.T.; Brown, M.G.; Jones, G.J.F.; Sparck Jones, K.; and Young, S.J. 1995. Video mail retrieval by voice: towards intelligent retrieval and browsing of multimedia documents. In IMMI 95.

Frohlich, D.M. 1991. The design space of interfaces. In Kjelldahl, L. (ed.), Multimedia principles, systems and applications. Berlin: Springer Verlag.

Hare, M.; Doubleday, A., Bennett, I.; and Ryan, M. 1995. Intelligent presentation of information retrieved from heterogeneous multimedia databases. In IMMI 95.

Huls, C. and Bos, E. 1995. Studies into full integratio of language and action. In Proceedings of the International Conference on Cooperative Multimodal Communication (CMC/95). part II. pp 161-174. Eindhoven, may 24-26.

IMMI 1995. Pre-Proceedings of the First International Workshop on Intelligence and Multimodality in Multimedia Interfaces: Research and Applications. Edited by John Lee. University of Edinburgh, Scotland, july 13-14.

Inder, R.; Oberlander, J.; and Tobin, R. 1995. Intelligent support for navigation in hypermedia: discourse structure and the Web. In IMMI 95.

Jackendoff, R. 1987. On beyond zebra: the relation between linguistic and visual information. *Cognition* 26(2):89-114.

Karagiannidis, C.; Koumpis, A.; and Stephanidis, C. 1995. Media/modalities allocation in intelligent multimedia user interfaces: towards a theory of media and modalities. In IMMI 95.

Krus, M. 1995. http://www.limsi.fr/Individu/krus/ and http://pmm-www.limsi.fr/~emux/

Martin, J.C. 1995. Cooperation between modalities and binding through synchrony in multimodal interfaces. PhD Thesis. In French. http://www.limsi.fr/Individu/martin/.

Martin, J.C. and Béroule, D. 1993. Types and goals of cooperation between modalities. In Proceedings of the 5th Conf. on Human-Computer Interaction (IHM'93), pp 17-22, 19-20 october, Lyon, France. In French. http://www.limsi.fr/Individu/martin/.

Martin, J.C.; Veldman, R.; and Béroule, D. 1995. Towards Adequate Representation Technologies for Multimodal Interfaces. In Proceedings of the International Conference on Cooperative Multimodal Communication (CMC/95). part II.. Eindhoven, may 24-26.

Maybury, M. 1991. Introduction. In Intelligent multimedia interfaces. AAAI Press. Cambridge Mass.

Nigay, L. and Coutaz, J. 1993. A design space for multimodal systems : concurent processing and data fusion. Proc. of INTERCHI'93. Amsterdam, april 24-29, 1993, ACM Press. pp 172-178.

Nigay, L. and Coutaz, J. 1995. Multifeature systems: from HCI properties to software design. In IMMI 95.

O'Nuallain, S. and Smith, A.G. 1994. An investigation into the common semantics of language and vision. *Artificial Intelligence Review* 8 (2-3):113-122.

Olivier, P. and Tsujii, J.I. 1994. Quantitative perceptual representation of prepositional semantics. *Artificial Intelligence Review* 8 (2-3).

Roques, M. 1994. Dynamic Grammatical Representations in Guided Propagation Networks. In Lecture Notes in Artificial Intelligence 862, R. C. Carrasco, J. Oncina (eds.) Grammatical Inference and Applications, pp 189-202, Second Intern. Colloqium, ICGI-94, Alicante, Spain, september 1994.

Salisbury, M.W.; Hendrickson, J.H.; Lammers, T.L.; Fu, C.; and Moody, S.A. 1990. Talk and draw: bundling speech and graphics. *IEEE Computer*., 23, 8, pp 59-65.

Santana, S. and Pineda, L.A. 1995. Producing coordinated natural language and graphical explanations in the context of a geometric problem-solving task. In IMMI 95.

Sims, R. and Hedberg, J. 1995. Dimensions of learner control: a reappraisal of interactive multimedia instruction. In IMMI 95.

Siroux, J.; Guyomard, M.; Multon, F.; and Remondeau, C. 1995. Oral and gesture activities of the users in the GEORAL system. In IMMI 95.

Väänänen, K. 1995. Four pillars for improving the quality of multimedia applications. In Proceedings of the first international workshop on evaluation methods and quality criteria for multimedia applications. november 4, San Francisco.

Vo, M. T. and Waibel, A. 1993. Multimodal Human-Computer Interaction. In Proceedings of the International Symposium on Spoken Dialogue: New Directions in Human and Man-Machine Communication, pp 95- 101, november 10-12, 1993, Tokyo, Japan.

Wahlster, W.; André, E.; Finkler, W.; Profitlich, H.J.; and Rist, T. 1991. Plan-based integration of Natural Language and Graphics generation. *AI Journal* 63:387-427.

Wang, E.; Shahnvaz, H.; Hedman, L.; Papadopoulos, K.; and Watkinson, N. 1993. A usability evaluation of text and speech redundant help messages on a reader interface. In G. Salvendy & M. Smith (Eds.), Human-Computer Interaction: Software and Hardware Interfaces. pp 724-729.

Westerlund, P.; Béroule, D.; and Roques, M. 1994. Experiments of Robust Parsing using a Guided Propagation Network. In Proceedings of the Intern. Conference on New Methods in Language Processing (NEMLAP), september 14-16, Manchester.