MASTER IN COMPUTER SCIENCE

Modeling and resolving conflicts and apprehensions in multimodal models

Pirau, Jonathan

Award date:
2016

Awarding institution:
University of Namur

Université de Namur
Faculty of Computer Science
Academic Year 2015–2016

Modeling and resolving conflicts and apprehensions in multimodal models

PIRAU Jonathan

UNIVERSITÉ DE NAMUR

Internship mentor: LALANNE DENIS

Supervisor: ______________________ (Signed for Release Approval - Study Rules art. 40)
DUMAS Bruno

A thesis submitted in the partial fulfillment of the requirements
for the degree of Master of Computer Science at the Université of Namur

Université de Namur
Faculté d'informatique
Année académique 20XX–20YY

MonTitre

Auteur1

Maître de stage : MonMaitredeStage

Promoteur : ________________________ (Signature pour approbation du dépôt - REE art. 40)
MonPromoteur

Co-promoteur : MonEventuelCoPromoteur

Confidentialité du mémoire:

Mémoire présenté en vue de l'obtention du grade de
Master en Sciences Informatiques.

Contents

Abstract

L'interaction multimodale permet à l'utilisateur de communiquer avec un ordinateur en utilisant différentes modalités et met en évidence la question de la fusion entre ces différentes modalités. Au début des années 1990, de nombreux travaux ont été réalisés donnant naissance à différent modèles de fusion multimodale. Ce travail réalise une analyse de ces modèles sur base des articles publiés à leur sujet, ainsi que les différents systèmes basés sur ces modèles. Cela permet d'en déterminer les forces et faiblesses et de déterminer la façon dont ces modèles interagissent avec l'utilisateur et le système. Suivant cette analyse une solution sera proposée : le modèle MUSE. Cette solution sera appuyée par des développements d'applications visant à l'exploiter au maximum ainsi que d'une évaluation auprès des utilisateurs. Cette étude est une comparaison de la compréhension de MUSE avec les modèles principaux de multimodalité tels que le CASE et le CARE. Ensuite un ajustement de la solution sera réalisé en suivant les résultats de l'évaluation ansi que du feeback des utilisateurs.
Mots clés : Multimodalité, CASE, CARE, TYCOON, systèmes multimodaux, utilisabilité, fusion de données, MUSE.

The multimodal interaction allows the user to communicate with a computer using different modalities. It bring to light the question of the fusion between those modalities. Back in the 90's, numerous work have been done on multimodal fusion models. This works analyze all those models using the papers published on them. The systems using those models are also analyzed. This allows us to determine what is good and bad in those models and how the interaction with the user and the system is managed. Following this, a solution will be proposed : the MUSE model. This solution will be supported by some application developments aiming to demonstrate all those properties and by a user evaluation. This user evaluation is a comparison of the understanding of MUSE and the main multimodals models such as the CASE and CARE. Then an adjustment will be done following the evaluation results and the user feedback.
Keywords : Multimodality, CASE, CARE, TYCOON, multimodal systems, usability, data fusion, MUSE.

Avant-propos

For the accomplishment of this work, I want first to thank Denis Lalanne and the whole HUMAN-IST group for the good welcome and supervision they gave me there in Switzerland during my work. I would thank Jacqueline and Bastien Ducret, the host family where I stayed those 3 months of internship. I want to thanks my family who has always been supportive to me in my studies, especially my godfather who has helped me a lot. Lastly I want to thank my sponsor, B. Dumas for giving me the opportunity to study this field of computer science.

Chapter 1

Introduction

In the last years, with the development of the devices for movement and speech recognition, first in the gaming industry, then in the general public, with the development of sensors and ambient intelligence, our way of communicating with the computers has smoothly changed using these devices in addition of classical ways to use the systems. But if it comes to light nowadays with the support of big companies such as Microsoft, Nintendo and so on, the first studies about this type of communication, called multimodal communication when a user uses multiple input channels to enter commands in a system come from the 80's. Then, in the 90's, there was a great research endeavor to model this, especially in France, leading in the development of multiple multimodality models. Afterwards the subject was a bit forgotten and now there is a great hype back around multimodal systems with the development of new advanced interfaces such as the KINECT, WiiMote and so on. This encourage us to look at the theoretical aspect of multimodality to see what the models are and if they are still relevant today, if they look complete and correct to the people who use them. The underlining research question of this work is "are the current multimodal models still reliable, complete and understandable". If the answer is negative, we will try to find a solution or propose an alternative.

In order to do this, we will use the following methodology, first we will make a state of the art to reference and analyze all the existing models in details and have a strong understanding of the multimodality field. We will take a deep look at the properties of those models, illustrate them and look for potential problems.
Secondly, we will look at different multimodal systems, seeing which model is used and how they organize the properties in application, if some are not used or if some used in all of them. We will begin with the reference systems of the models in the state of the art then we will take a look at other laboratory projects and commercial developments.

Thirdly, if problems are detected in the models, or if the systems study shows that they lack something, we will try to propose a solution based on the models if possible. This solution will be detailed and compared to the other models, showing what it can bring new. Some tests systems will be developed to help us reasoning and validating the model.
Fourthly, this new solution should be evaluated with the users to analyze if the supposed added value to the other models is real and understandable by the users.
Finally, if necessary we will improve our solution using the result of this evaluation, trying to have the most complete, consistent and understandable multimodal model.

During the realization of this work, an internship has been made at the University of Fribourg, in the HUMAN-IST research group of the department of computer science of the science faculty with the professor D. Lalanne from the 14th of September to the 15th of December 2015. This work is also supervised by the professor B. Dumas from the University of Namur. The principal research tool for the sources of this work will be Google Scholar for its facility to find paper on the different libraries. The used libraries will be ACM digital library, IEEExplore, researchgate, springer, citeseer with the easy access provided by the University of Fribourg.

Chapter 2

State of the art

2.1 What is multimodality?

Multimodality allows the user to employ different channels of communication (called modalities) to interact with the computer. Those channels can be very different, starting from the keyboard and mouse to the pitch intensity and emotion recognition, passing by the speech and gesture recognition. If humans are inherently multimodal [12], the interaction with a computer is often limited to the mouse and the keyboard, but there are many other ways to communicate with a computer : this is what multimodality is doing.

2.2 Advantages of multimodality

Even if an enhanced efficiency seems to be the main advantage of the multimodal systems, Sharon Oviatt shown that the efficiency is only increased by 10 percent compared to unimodal commands [29]. The biggest advantages versus a classical system are less errors, enhanced usability, flexibility and mutual disambiguation [29].

2.3 "Put-That-There"

The "Put-That-There" from Richard A. Bolt [2] is referenced as the genesis of the multimodality. In that system, the user is sitting on a chair with joysticks, in front of a big white screen, interacting with the system with the joysticks, some buttons, and voice command [2]. The Figure 2.1 represents the sketch of the media room.

The system in itself is a drawing editor, where the user can draw shapes of color, the famous put-that-there command results when some shapes are draw on the editor, and the user wants to move one of them. But this was not the only possible command, in addition to the drawings ones such as "Draw a green circle here" there were a few other examples [2] :

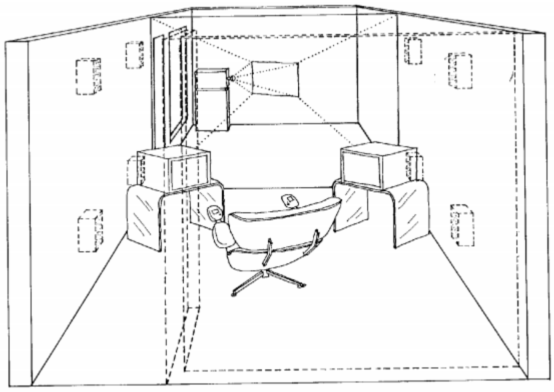


Figure 2.1: Sketch of the Bolt's system media room [2]

- "Move that to the right of the green square"
- "Make that smaller"
- "Make that a large blue diamond"
- "Name that ... the calendar"
- "Delete the large blue circle"
- "Delete that"

The user sometimes makes a pause after a deictic reference to let time to the system to blink the indicated object, which means that the item is located and recognized. The Figure 2.2 shows R. Bolt working with the system.
Even though the system was probably a fake (some people say that someone was hiding and executing the commands manually, others say that the recognition worked, maybe you can make your opinion there :
https://www.youtube.com/watch?v=0Pr2KIPQOKE [15] but this is not the point here), it has shown the advantages and the usability of multimodality. R. Bolt says that it shows the versatility and ease of use that the management of graphic space with voice and gesture can bring [2]. He also says that the manipulations are more spontaneous and natural with this system working with spoken words than the ones working with typed symbols [2].

2.4 CASE

Presented in the INTERCHI 93 conference in April 1993 by Laurence Nigay and Joëlle Coutaz, the CASE model is the first referenced model of multimodality fusion. It describes the way how commands are interpreted by a

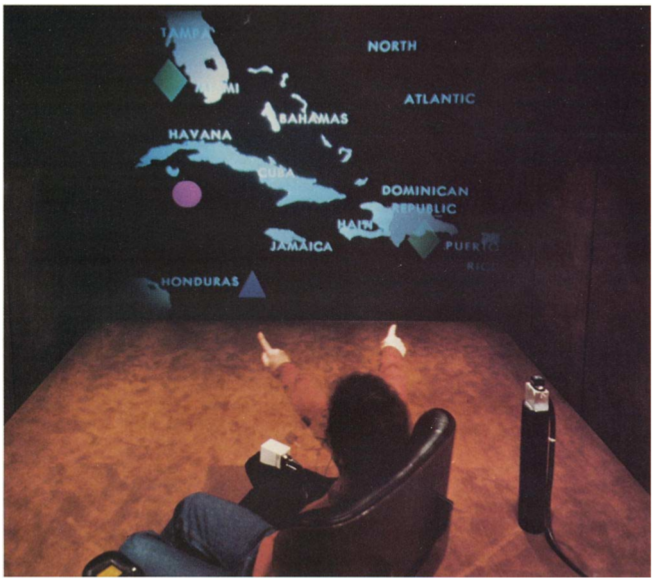


Figure 2.2: Interface of the Bolt system [2]

system showing four basic properties for the fusion of input modalities, represented in the Figure 2.3. The modalities can be combined in two different ways : if the use is sequential or parallel, and combined or independent.

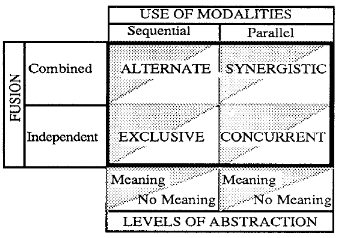


Figure 2.3: Properties of CASE model [28]

This results in four properties which will be analyzed below :

- Concurrent
- Alternate
- Synergistic
- Exclusive

2.4.1 Concurrent

In the concurrent property, the modalities are used in the same time but with no co-reference. The Figure 2.4 shows a graphical representation of the concurrent property.
Example : Within VoiceFinder, a system that adds voice input to the MacIntosh Finder , the user can issue a voice command like "empty the trash" while simultaneously invoking another command such as opening a document with the mouse [28].

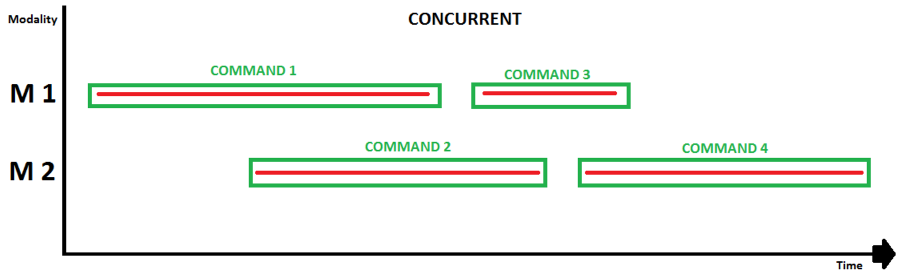


Figure 2.4: Concurrent property

2.4.2 Alternate

In the alternate property, the modalities are used in a sequential order in a combined way. One modality at a time, multiple modalities making a command. The Figure 2.5 shows a graphical representation of the alternate property.
For example, in a system where the interaction is driven by natural written language, the deictic references that may occur in a sentence with a word like "this", are solved by looking for mouse selections in the next following act of interaction: modalities are combined but acquired in a sequential order. [28]

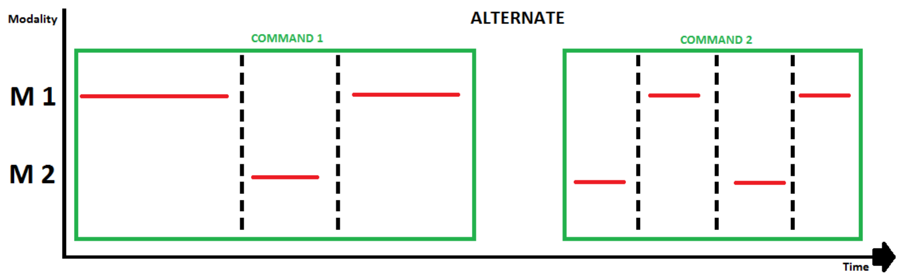


Figure 2.5: Alternate property

2.4.3 Synergistic

In the synergistic property, modalities are used in parallel and a combined way. One command requires multiple modalities. The Figure 2.6 shows a graphical representation of the synergistic property.
For example : "Insert a note" is a synergistic command: it is specified using speech and mouse click simultaneously and it requires the fusion of data from multiple input devices. [28]

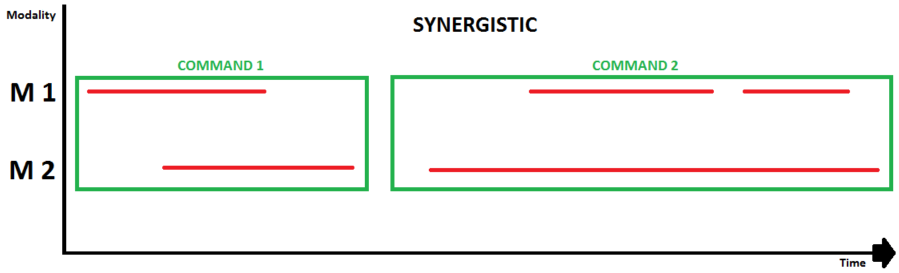


Figure 2.6: Synergistic property

2.4.4 Exclusive

In the Exclusive property, modalities are independent, and used in a sequential order, one modality per command, no overlapping between modalities, one command at a time. The Figure 2.7 shows a graphical representation of the alternate property.
Example : When editing the content of a note only one modality is available (typing) and no other command can be invoked in parallel. For example, it is not possible to turn the pages of the NoteBook (see further) while writing the content of a note. [28]

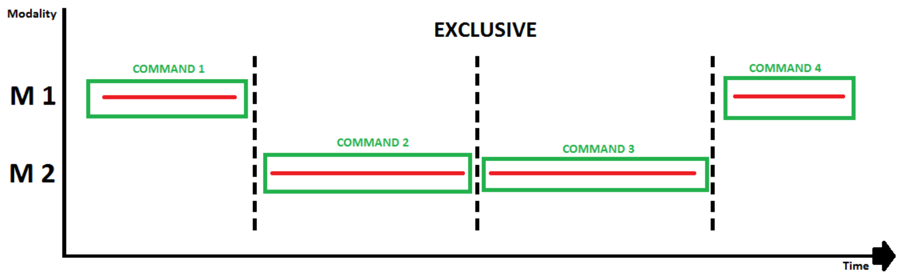


Figure 2.7: Exclusive property

2.4.5 Systems using the CASE properties

In their paper, J. Coutaz and L. Nigay show two multimodal applications following the CASE model [28] : VoicePaint and NoteBook.
VoicePaint is a graphic editor implemented on a Macintosh using a word-based speech recognizer called Voice Navigator. The user can draw with the mouse and use his voice to change some parameters like the color of the pen, its thickness [28]. The paper did not explain if a given command can be entered with multiple different modalities, and furthermore, what is happening if there is a redundancy between those modalities.
NoteBook is a personal electronic book implemented on the NeXT machine using a continuous multilocutor speech recognition system called Sphinx. The user can create, edit, browse, and delete textual notes. For example, the user can insert a note between two others by pointing the location between the two notes and saying "Insert a note here" [28]. Unlike the VoicePaint, the paper tells that the system allows multiple possible modalities for a command : to empty the notebook, the user can utter "Clear notebook" or click on the button "Clear", but nothing is explained about the result of entering the two commands simultaneously.

2.5 CARE

Following the CASE and presented by the same research group in 1995, the CARE model focuses more on the multimodal system state. The properties are represented by state machines and show how the user can interact with them. Like the CASE model, it gives four properties of multimodal systems :

- Complementarity
- Assignment
- Redundancy
- Equivalence

The formal expression of the CARE properties relies on the notions of state, goal, modality and temporal relationships. I will not present those properties in the order they are given because they are interdependent, some are required by the others.

2.5.1 Complementarity

For the complementarity property, multiple modalities are required during the temporal window to go from a state S to a state S', none of them can do it individually. The Figure 2.8 shows a graphical representation of the

complementarity property.
Example : Flight to + pointing a destination [9]

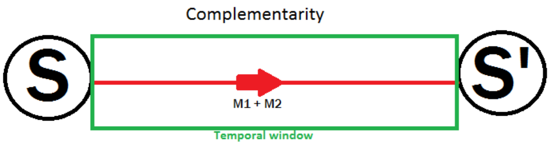


Figure 2.8: complementarity property

2.5.2 Equivalence

Multiple modalities permit to go from the state S to the state S' but one is enough to make the transition.
Example : specifying "Pittsburgh" as the destination of a trip. Users have a choice of speaking or typing the sentence "Flights to Pittsburgh", or keying "Pittsburgh" in the destination slot of the request form [9] The Figure 2.9 shows a graphical representation of the strict assignment property.

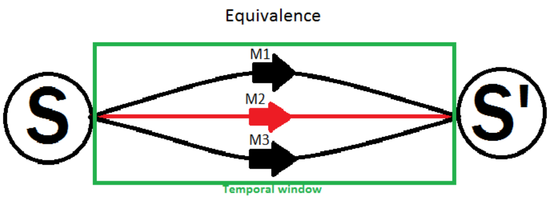


Figure 2.9: Equivalence property

2.5.3 Assignment

In the assignment, only one modality is required to go from a state S to a state S' and no other modality is used.
There are two types of assignment : the strict Assignment and the agent Assignment.

Strict Assignment

Only one modality is allowed by the system to go from a state S to a state S'. The Figure 2.10 shows a graphical representation of the strict assignment property.
Example : window manipulation is only possible by direct manipulation only. [9]

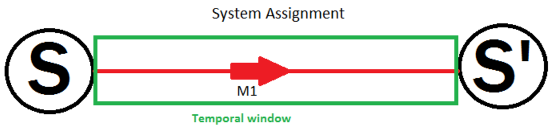


Figure 2.10: Strict Assignment property

Agent Assignment

Multiple modalities are allowed to go from a state S to a state S' but the user always uses the same. The Figure 2.11 shows a graphical representation of the agent assignment property. This property can't exist without equivalence.
Example : a user who always specify his destinations by speech [9]

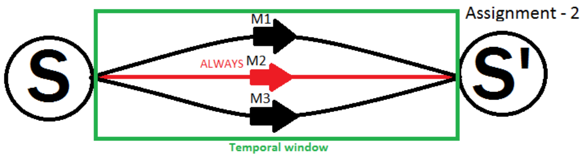


Figure 2.11: Agent Assignment property

2.5.4 Redundancy

Multiple modalities permit to go to S' from S, one is enough, but the agent takes multiple of it. The Figure 2.12 shows a graphical representation of the redundancy property. Same as the agent assignment, this property can't exist without equivalence.
Example :"flight to Pittsburgh" + pointing it [9]

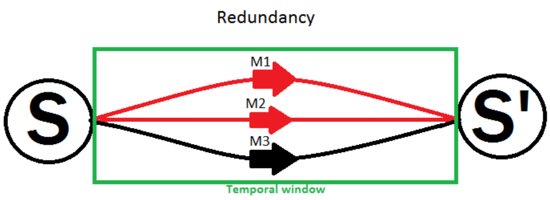


Figure 2.12: Redundancy property

2.5.5 System using the CARE properties

In the presentation paper FOUR EASY PIECES FOR ASSESSING THE USABILITY OF MULTIMODAL INTERACTION: THE CARE PROPERTIES [9], the CARE model is presented alongside the MATIS system

(Multimodal Airline Travel Information System). In this system, the user is allowed to retrieve information about flight schedules using, the keyboard and voice modalities. The system is supporting individual and synergistic use of the input modalities [9].
The problem here is that the system only handles synergistic use of the modalities, but what about the concurrent one ? Specifying something by voice while doing something else with the keyboard and mouse.

One other interesting thing to note is that the only example of strict assignment in MATIS is the window manipulation, all other functionalities have equivalence, when there are plenty of examples of agent assignment.

2.6 TYCOON

The timeline of the TYCOON model is quite anarchic. In an article over the design space for multimodal interaction in 2004 [27], Laurence Nigay speaks about it in a seminar in 1997 presenting the TYCOON and six types of cooperation. In 1998, J-C Martin published the most cited article over the TYCOON [22] but with only five properties. One year later, in 1999, a new article over the TYCOON shows up [23], taking the six original properties, but this article is less noticed than the first.
The TYCOON aims to complete the CARE model, and adding facilities enabling a faster multimodal interaction. For that it takes three properties of the CARE model, transforms the fourth and adds two new ones.
The properties of the TYCOON are :

- Complementarity
- Equivalence
- Redundancy
- Specialization
- Transfer
- Concurrency

The Complementarity, Equivalence and Redundancy will not be explained here since they are exactly the same as in the CARE model explained above.

2.6.1 Specialization

When modalities cooperate by specialization, this means that a specific kind of information is always processed by the same modality [22]. The difference between previous properties is that we are not in state machine properties

anymore, we are in a relation between events and modalities.
There are three kinds of specialization :

- Data relative specialization
- Modality relative specialization
- Absolute specialization

Data relative specialization

We speak about data-relative specialization if an event only uses one modality to communicate with the user [22].
The Figure 2.13 shows a graphical representation of that property, showing an event E1 which always uses the modality M2.

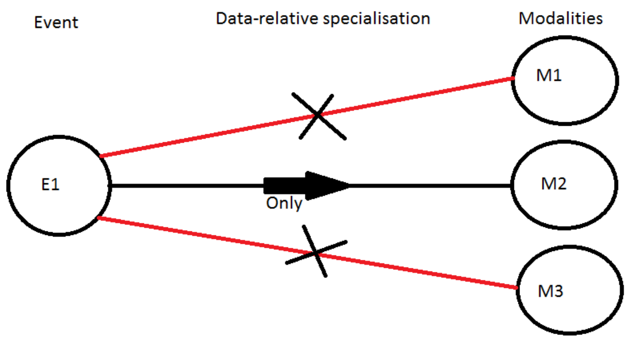


Figure 2.13: Data-relative specialization property

In output modalities it is a data-relative specialization if errors only produce sounds and no graphics or text [22].
Another example occurs when the destination of the speech command "Fly to" can only be typed and the typing can be used to enter other commands too.
As we can see, this is the same as the system assignment from the CASE model.

Modality relative specialization

We speak about modality-relative specialization if a modality is only used by one type of event [22].
The Figure 2.14 shows a graphical representation of that property, showing a modality M1 which is only used by the event E2.

In output modalities it is a modality-relative specialization if sounds are not used to convey any other type of knowledge [22].
Another example is when typing is only used to specify the destination of

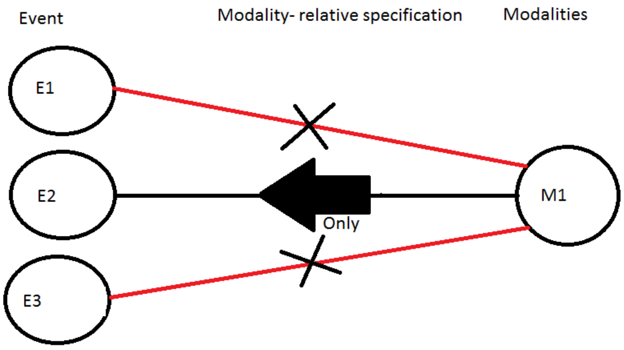


Figure 2.14: Modality-relative specialization property

the speech command "Fly to", but the destination can be specified by speech too.

Absolute specialization

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 [22]. An example is when the destination of the speech command "Fly to" can only be typed and this is the only event when typing is allowed.
The Figure 2.15 shows a graphical representation of that property, showing a modality M2 who is only used by the event E2 and the event E2 which always use the modality M2.

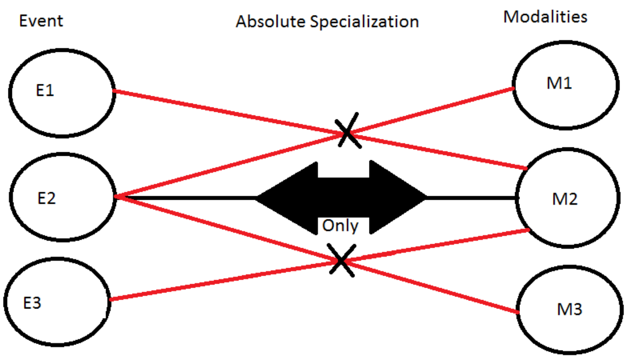


Figure 2.15: Absolute specialization property

2.6.2 Transfer

The transfer property is quite different from the previous ones since most of the time it involves a computation of a modality to add information from another. Here is the definition from J.-C. Martin : when several modalities cooperate by transfer this means that a chunk of information produced by a modality is used by another modality [22].
The most meaningful example is : 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 whole sentence [22].

2.6.3 Concurrency

The last property is the Concurrency, added in the paper over the TYCOON in 1999. A cooperation by concurrency means that several agents produce independant chunks of information at the same time. These chunks must not be merged. [23]
The Figure 2.16 shows a state machine representing this property. Here is a situation that the state machine can possibly represent : we have a drawing editor, in which you can enter the commands by the voice or mouse. We are on the state S, representing the white sheet. We want to do two things : drawing a red circle and a green square. We can use one modality to do one of those tasks at a time, going to the states S' and S(rond) from S , and then going to the state SF. Or we can draw the circle with one modality and the square with another at the same time using concurrency, and go directly from S to SF using only one temporal window.

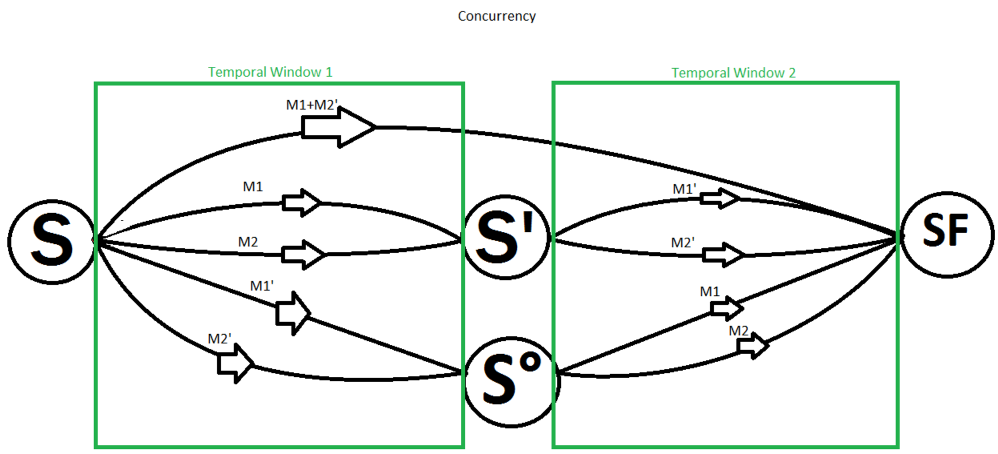


Figure 2.16: Concurrency property

Chapter 3

System examples

3.1 Case study : RASA military system

3.1.1 Introduction

The RASA project is a case study on how an augmented multimodal environment can support work practices and enhance them with digitization. The augmented work practice in the RASA system is a military command post installed in a war zone. One responsibility of officers is to track the movement of friends, enemies, and neutral parties and to report it on a map. Before the project, they used a kit of useful items from everyday objects like a high-fidelity paper map of the terrain, some way to hang the map, objects used to symbolize the units and pens. Surprisingly, the most often used object to represent the units is a Post-It with some symbols on it.
The command post keeps the Etat-Major and the other command posts updated by radio. It is the job of some officers to keep all the information on the map updated, so their superiors can make critical decisions quickly. The advantage of digitization are numerous : the first is that the other command posts are updated in real-time and the Etat-Major can follow what happens in the posts easily. All the data are centralized and all the users can have a clear picture of the current state in one look at the map. [24] The Figure 3.1 represents the situation in one of those command posts (this picture comes from the original RASA paper [24]).

3.1.2 Constraints

There are numerous constraints: the change to the work practices should be minimal, in a combat zone, soldiers do not have the time to think "How do I have to use the system ?" The system should support numerous end-users, all the information should be modifiable because it is gained over time, the users must be able to understand their own augmentations (even when the system is not working, the user should be able to have the information about

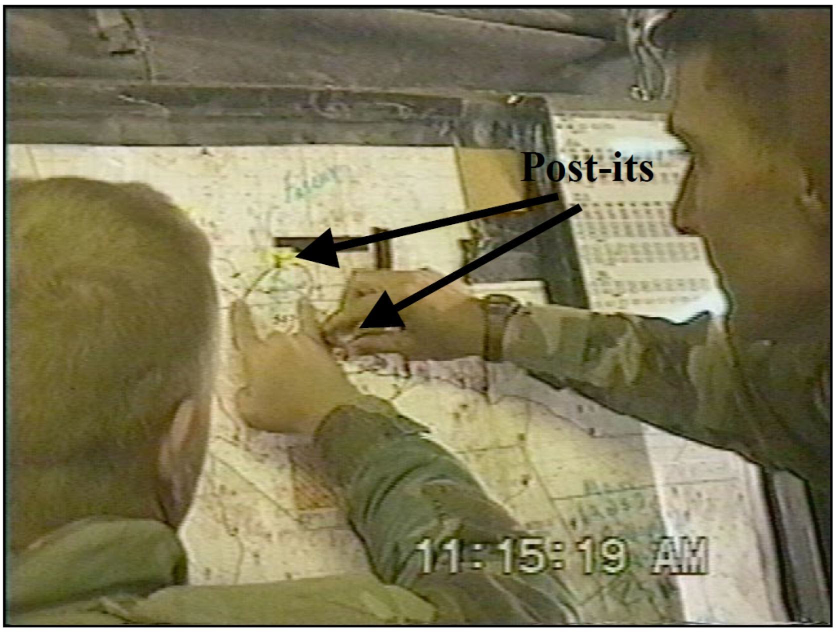


Figure 3.1: actual map and overlay in a command post

the current state just by looking at it and having and good understanding of what the used symbols means). Finally the system should be extremely robust : It should work in really bad weather conditions like in the desert and the digital equipment should be minimal. A map with a bullet hole is still usable when a computer with a bullet hole is basically a rock.

3.1.3 System presentation

The system is composed of a smartboard where a map can be attached (the map needs to be localized in the virtual representation of the world), Post-its, and an ink and speech recognizer. Here is how the system works (figure 3.2, this image comes from the original RASA's article [24]).

The basic use-case of RASA is the identification and position of enemy units on the map (this use case is the one presented in the RASA paper): a soldier draws the unit symbol used in Figure 3.1 and at the same time says the unit's name, for example "Romeo-One-Bravo". Next to this the user places the Post-it on a registered map of the terrain in response to which RASA says : "Confirm? Enemy mechanized regiment called Romeo-One-Bravo has been sighted at nine-six, nine-four". [24] This basic use case is really interesting, let's take a look at the timeline of the modalities (fig 3.3).

How this use case places itself on the multimodality models?

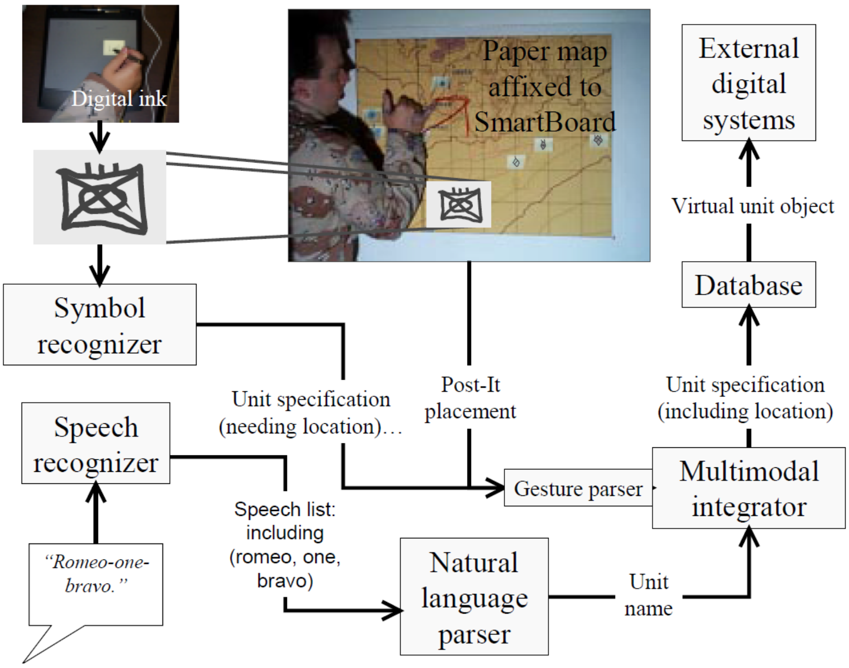


Figure 3.2: Workflow of RASA

For the TYCOON and CARE, the answer is obvious, it is one of the perfect examples of the Complementarity property.
For the CASE, it is more difficult, the modalities 1 and 2 are combined in a Synergistic way, and this combination is combined with the third modality using the Alternate property. In the model, the commands are defined by only one property, a hybrid case is never presented, so this command cannot be strictly placed on the model, you need to extrapolate it a bit.
One of the constraints of the system is that it should support numerous end-users, so we can imagine a use case like this : two soldiers in front of the system, one entering a command as in the previous example, the second drawing circles and labeling them (with labels such as mines, no-go areas, etc). This time the timeline is like this (3.4) :

We have two possibilities to analyze this use case : user by user or globally, we will take a look at both. The table 3.1 shows the properties handled by RASA.

User by user : (fig 3.4)

For the user one, it is the same as the previous example. The second user is communicating in a unimodal way with a multimodal system. It is inter-

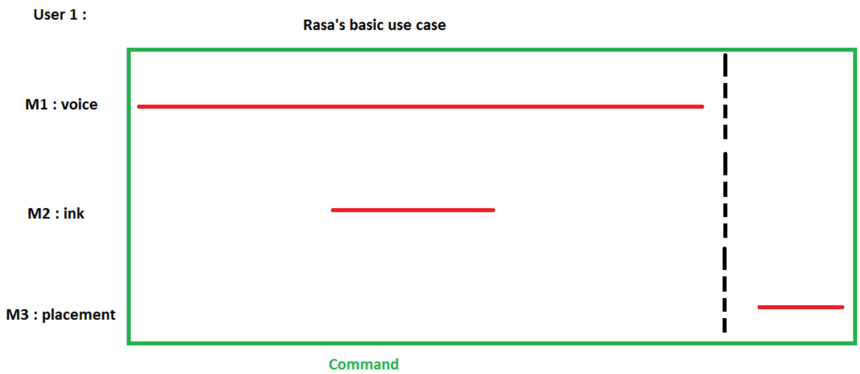


Figure 3.3: RASA's basic use case

Table 3.1: Multimodal properties handled by RASA

CASE		CARE		TYCOON	
Concurrent	OK	Complementarity	OK	Complementarity	OK
Alternate	OK	Assignment	KO	Equivalence	OK
Synergistic	OK	Redundancy	OK	Redundancy	OK
Exclusive	KO	Equivalence	OK	Specialization	?KO
				Transfer	?
				Concurrency	OK

esting to see how the models represent that concept with the "nuancy" they bring. In the CASE what we just described corresponds to the Exclusive property while in the CARE this corresponds to the Assignment property and more specifically the Agent Assignment property since the system offers multiple ways to enter the command and the user is only using ink (even if it is because the other modality is busy).
This example also meets the definition of the equivalence property which is used in both CARE and TYCOON. In the TYCOON model, we can think that we have specialization. Data or modality specialization ? The ink has numerous uses in the system (icons, zone delimitation, arrows, labels,...) so this is not modality specialization. The minefields are identified by ink zones and ink labels in this example, but labels can also be given by speech, so this is neither data specialization.

Globally (fig 3.5) :

When we take this two-users simulation globally, we still have 4 modalities if we make the distinction between the 2 input flows of ink.
Placing this use case on the CARE model is relatively easy, it is Comple-

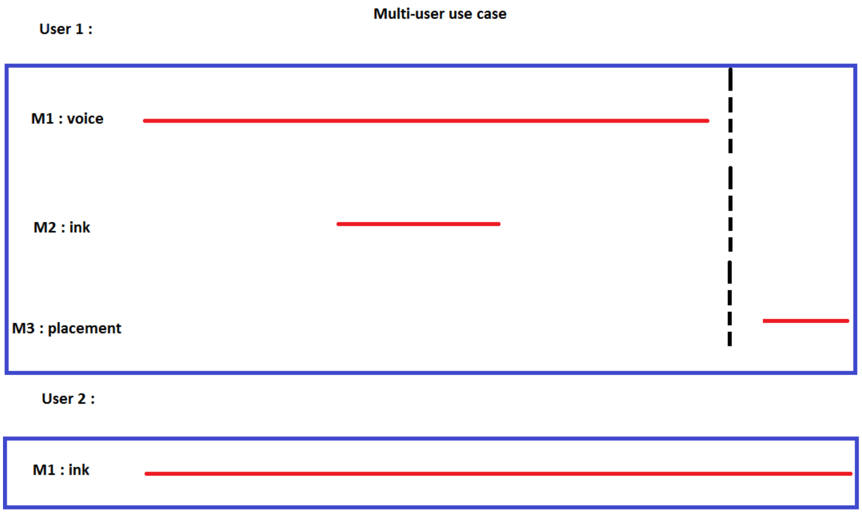


Figure 3.4: RASA multi-user use case, user separated

mentarity, but reading the example, we have the feeling that it is a bit more than this. On the CASE, it becomes tricky. It is clearly a synergistic command, but if you look closer, you can find the alternate property too (M1 and M3 - M2 and M3). In addition, since the two soldiers work on the same schema but are doing different tasks, it fit the Concurrency definition well too. In the TYCOON model, the two properties encountered are of the types : Complementarity and Concurrency.

Conclusion

Multi-user systems are a bit more difficult to analyze with the classical models, you have to apply them to each of the users and to the global situation due to the mix of systems reserved properties such as equivalence and user reserved properties such as assignment. In addition, we have the feeling that the CARE lacks some properties (like the Concurrency) and the CASE too (Redundancy). The TYCOON seems a bit more complete but lacks of a property definition for someone using a multimodal system with only one modality since the model is a kind of system-focused CARE.

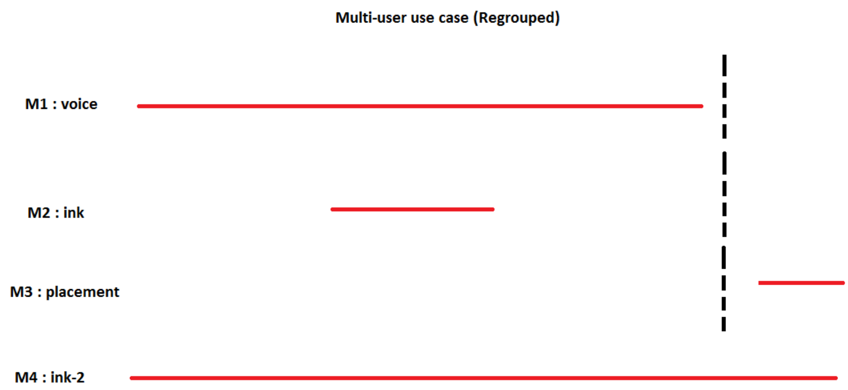


Figure 3.5: RASA multi-user use case, user regrouped

3.2 Analyse of example systems with the CASE-CARE-TYCOON models

3.2.1 MAICO [35]

The goal of the MAICO project is to observe how the users can interact with a multimodal system and more precisely how the system responses can influence the user behavior. To achieve this goal, they have studied how humans exchange information, especially the nonverbal information. They believe that introducing this kind of communication on computer systems may change the behavior of the interfaces to make them more natural and attractive. Following those beliefs, they have made a human shaped agent to communicate with the user. She is called Multimodal Agent Interface for COmmunication or simply MAICO [35]. The Figure 2.6 shows a sample of MAICO's expressions.

MAICO is equipped with an Image Recognizer, Speech Feature Extractor, Keyword Spotter, Natural Language Processing, DDM, CG Agent Generator, and Speech Generator. The fig 3.6 shows a representation of how those parts are working together.

Now that we understand how the system works, let's see which properties of the different multimodal models it handles. This article does not contain many use case examples, when a property is handled there is an OK, when it is not there is a KO and if the paper does not contain enough information to determine it a "?" is indicated. This is shown in the Table 3.1.

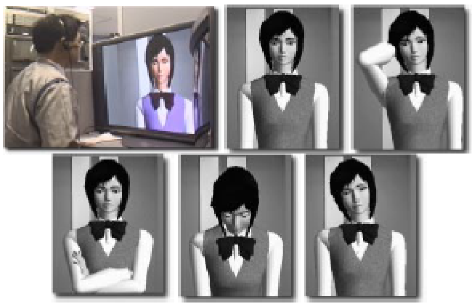


Figure 3.6: MAICO's actions (normal, at a loss, confusing, bow, question) [35]

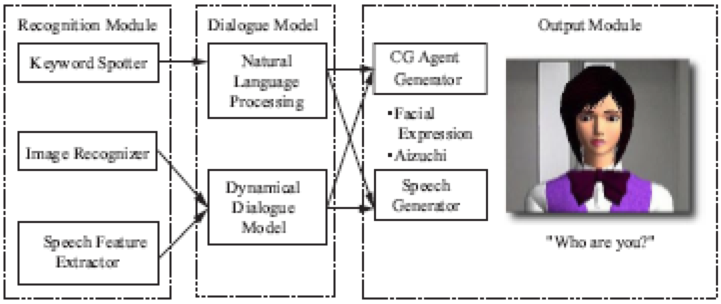


Figure 3.7: MAICO's workflow [35]

3.2.2 Multimodal Learning Interface [5]

A computer has to learn the tasks performed by a human every day, like unscrewing a jar, stapling a letter and pouring water. For that, the system watches the user doing the task and listens him describing what he is doing. Data are put in a fact database, so when the system sees the user doing the task again sometime later, he is able to recognize it. It is also able of giving the different subtasks of a task when asked. This type of self-learning systems can be used in smart home environments. The learning system is composed of [5] :

- Attention detection to follow the gaze and head movements.
- Attentional object spotting and action categorization with extraction of the non-spoken information such as attentional objects and intentional actions.
- Speech segmentation and word spotting
- Multimodal integration which performs the fusion between the 3 modalities.

Table 3.2: Multimodal properties handled by MAICO

CASE		CARE		TYCOON	
Concurrent	?	Complementarity	OK	Complementarity	OK
Alternate	?	Assignment	OK	Equivalence	OK
Synergistic	?	Redundancy	?	Redundancy	?
Exclusive	OK	Equivalence	OK	Specialization	OK
				Transfer	?
				Concurrency	?

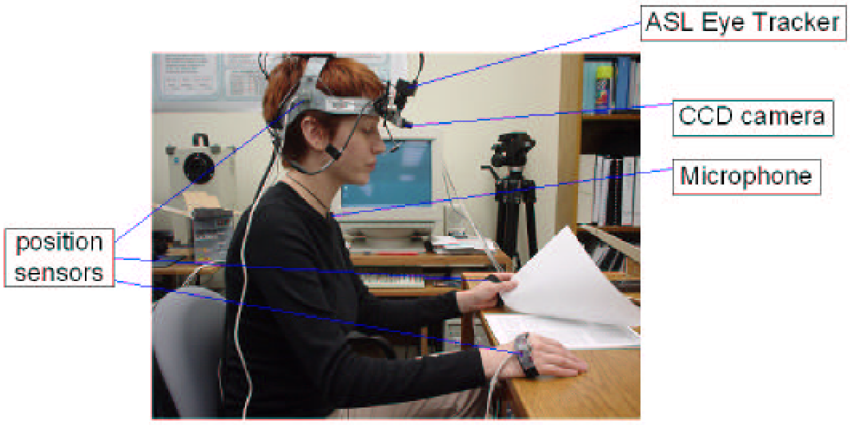


Figure 3.8: Description of the learning interface [5]

The table 3.2 shows the model properties handled by the multimodal learning interface, a "?KO" means that the text does not show any example proving that the property is not followed by the system, but that we can suppose it by the way the system works. The Figure 3.8 shows the position of the learning interface sensors on the users.

3.2.3 MIMI [33]

MIMI is a multimodal interface for in-car communication systems. Those systems become a common feature in nowadays cars, but they often require a lot of attention from the driver when this one had already many things to do. The goal of the MIMI project is to provide an in-car communication system where the driver's hands and eyes are solely allocated to the driving task. To achieve this MIMI uses a speech recognizer and some buttons on the driving wheel (see fig 3.9) [33].
The fig 3.10 shows examples of commands that can be entered to the system and the table 3.3 the properties followed by the system.

Table 3.3: Multimodal properties handled by the multimodal learning interface

CASE		**CARE**		**TYCOON**	
Concurrent	OK	Complementarity	OK	Complementarity	OK
Alternate	?OK	Assignment	?KO	Equivalence	OK
Synergistic	OK	Redundancy	OK	Redundancy	OK
Exclusive	?KO	Equivalence	OK	Specialization	?KO
				Transfer	?KO
				Concurrency	OK

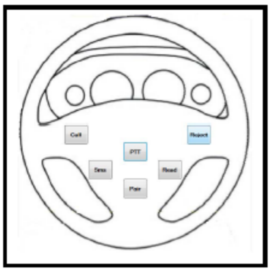


Figure 3.9: Buttons on the driver wheel [33]

3.2.4 MATCH [25]

MATCH is a multimodal mobile information access system. Running on a PDA (figure 3.11) MATCH allows you to gather information on restaurants or subways using speech and ink recognition. While walking in a city, the user can ask the system to find a nearly Italian restaurant, or the next subway to a fixed destination. The fig 3.12 shows an example of ink recognition. The table 3.4 shows the properties handled by MATCH.

3.2.5 MUMS [34]

The MUMS project is a bit like MATCH, it is an application running on a PDA, giving information to the user but this time it is more about navigation and public transportation, going from a point A to a point B. The typical example is "I want to get there from here" + pointing a location on the map. Like MATCH this project uses ink and speech recognizer but the ink is only used to indicate a location or an area, the command verbs should be uttered. The Figure 3.13 shows the interface of MUMS. The table 3.5 shows the properties handled by MUMS.

Task ID	Calling a contact
T01	Please call Patrick by saying "Call Patrick"
T02	Please call Felix by saying "Dial Felix"
T03	Please call Janet; Push the call button and say "Janet"
	Calling a dictated number
T04	Please call the number 0, 7, 7, 4, 4, 4, 5, 5, 5, 5 - Say "Call" or "Dial" - Dictate digit by digit 0, 7, 7, 4, 4, 4, 5, 5, 5, 5 - Press "C" = <Complete>
T05	Please call the number 0, 7, 2, 2, 2, 2, 3, 3, 3, 3 - Say "Call" or "Dial" - Dictate 0, 7, 2, triple 2, double 3, double 3 - Press "C" = <Complete>
T06	Please call the number 8, 8, 8, 1, 1, 1, 6, 6, 6, 6 - Say "Call" or "Dial" - Dictate 0, 7, 2, triple 2, double 3, double 3 - Press "C" = <Complete>
T07	Please call the number 0, 7, 7, 6, 6, 6, 2, 2, 5, 5 - Press "C"=<Call> - Dictate the number using your preferred method - Press "C" = <Complete>
	Other commands
T08	Please redial the last outgoing call by saying "Redial"
T09	Please call back the last incoming call by saying "Callback"
	Sending text message to a contact
T10	Please send a message to Patrick - Say "Send message to Patrick" - Choose the options "One"
T11	Please send a message to Andre - Say "Sms Andre" - Choose the options "Three"

Figure 3.10: Examples of MIMI commands [33]

3.2.6 Multimodal Dialogue Systems for Interactive TV Applications [1]

This system is a multimodal program guide for an interactive television. The user can ask questions to the system with the TV remote controller and speech instructions. A typical use case is the user asking the films on channel cinema between 9 pm and 8 am, the system responding with a list of movies then the user browses them with the remote controller to have additional information such as the actors starring in the film or the movie director. The system is always used in this way in the examples, one modality at a time, one modality per command. The Figure 3.14 shows a representation of the interface and the table 3.6 represents the properties handled by the multimodal TV application.

3.2.7 Conclusion

After RASA, we have analyzed six additional systems, the most represented property is the Equivalence, which is the heart of multimodality : giving the user multiple ways of performing a task. The second one is the Complementarity and its declination : Alternate and Synergistic. You cannot find one of these two in a system without the second one. The Assignment or

Table 3.4: Multimodal properties handled by MIMI

CASE		**CARE**		**TYCOON**	
Concurrent	KO	Complementarity	OK	Complementarity	OK
Alternate	OK	Assignment	OK	Equivalence	OK
Synergistic	OK	Redundancy	OK	Redundancy	OK
Exclusive	OK	Equivalence	OK	Specialization	OK
				Transfer	?KO
				Concurrency	KO

Figure 3.11: MATCH running on a PDA [25]

Exclusive property is present in nearly all systems too, but it is more a User Assignment than a System Assignment, which is really not represented, the systems aim to offer multiple possibilities. The Transfer is often present too, but it is because the systems often have only one output device which is a screen. The less represented property is Concurrency, it seems that doing multiple things at one time is not natural for those systems, but most of them address themselves to only one user at a time. The Specialization property is a bit unloved, the systems do not like to specialize their modalities in only one type or data, or data in only one modality. The motto of multimodality can be "Diversity".

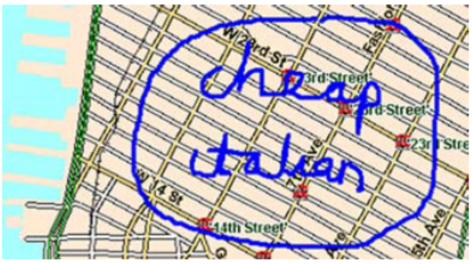


Figure 3.12: Example of ink recognition [25]

Table 3.5: Multimodal properties handled by MATCH

CASE		**CARE**		**TYCOON**	
Concurrent	KO	Complementarity	OK	Complementarity	OK
Alternate	OK	Assignment	OK	Equivalence	OK
Synergistic	OK	Redundancy	OK	Redundancy	OK
Exclusive	OK	Equivalence	OK	Specialization	KO
				Transfer	?OK
				Concurrency	KO

Figure 3.13: MUMS interface running on a PDA [34]

Table 3.6: Multimodal properties handled by MUMS

CASE		**CARE**		**TYCOON**	
Concurrent	KO	Complementarity	OK	Complementarity	OK
Alternate	OK	Assignment	OK	Equivalence	OK
Synergistic	OK	Redundancy	?	Redundancy	OK
Exclusive	OK	Equivalence	OK	Specialization	OK
				Transfer	?OK
				Concurrency	KO

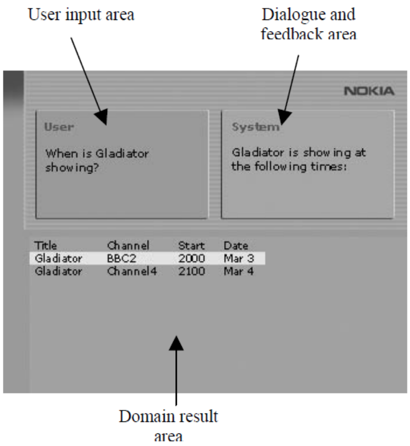


Figure 3.14: TV application interface [1]

Table 3.7: Multimodal properties handled by the multimodal TV application

CASE		**CARE**		**TYCOON**	
Concurrent	KO	Complementarity	KO	Complementarity	KO
Alternate	KO	Assignment	OK	Equivalence	OK
Synergistic	KO	Redundancy	KO	Redundancy	KO
Exclusive	OK	Equivalence	OK	Specialization	OK
				Transfer	?OK
				Concurrency	KO

Chapter 4

Properties reorganization

4.1 Ambiguities

In practice, the most used models are the CASE and the CARE, but we have demonstrated in the previous sections that those models are not enough. This was understood by the researchers because each of them makes his own interpretation of the models for his needs. The models are often misunderstood too, the CARE model is often associated with user properties, but we see in the previous section that all its properties are resumed to state machines, only the Agent Assignment and Redundancy are linked to the user, when the CASE is associated with the system. The organization between those models is a question too, are those models complementary or should one replaces the others ? For the CASE and the CARE, the models seem to overlap a bit, since the Synergistic and Alternate properties from the CASE can be placed in the Complementarity from the CARE, the CARE model is a bit like an evolved version of the CASE, but with some other aspects such as the user's role.
A preliminary study shows that the Transfer and Specialization properties from the TYCOON do not really convince the researchers because these properties need more than one period of time to be observed.

4.2 The goal

In multimodality, the dream is to have one unified model, or separated ones, but which treats different aspects, such as the user side and machine side, which can be clearly used by the developers to create multimodals systems.

4.3 Properties listing and organization

4.3.1 Properties listing

At this point, excluding the CASE, we can extract seven properties from the main models (CARE-TYCOON) :

- Conplementarity
- Equivalence
- Redundancy
- Concurrency
- Transfer
- Specialisation (Data - Modality - Absolute)
- Assignment (User - System)

As we already said before System assignment and Data specialization have the same definition.

4.3.2 Primary organization

We can see that we have different types of properties here :

- Properties on the system state, how the system can go from a state to another (Complementarity - Equivalence - System Assignment).
- Modality specialization : express the reserved use of a modality for conveying a special type of knowledge (is it really useful in input ?).
- Properties of how the user can interact with the system :
  - Using one modality at a time (User assignment - Unimodality).
  - Using multiple modalities at a time (Redundancy - Concurrency - Complementarity).
- Transfer - a modality is converted into another.

4.3.3 Placement on Dumas and Lalanne's modified Norman action circle [3]

Now, we can place those properties with the CASE model on the Human action cycle modified by Dumas and Lalanne based on Norman's Action Cycle and Nigay's Pipe-Line Model [3]. This is shown in Figure 2.1. Modality Specialization has no real sense in input, so it's put on the output, and at

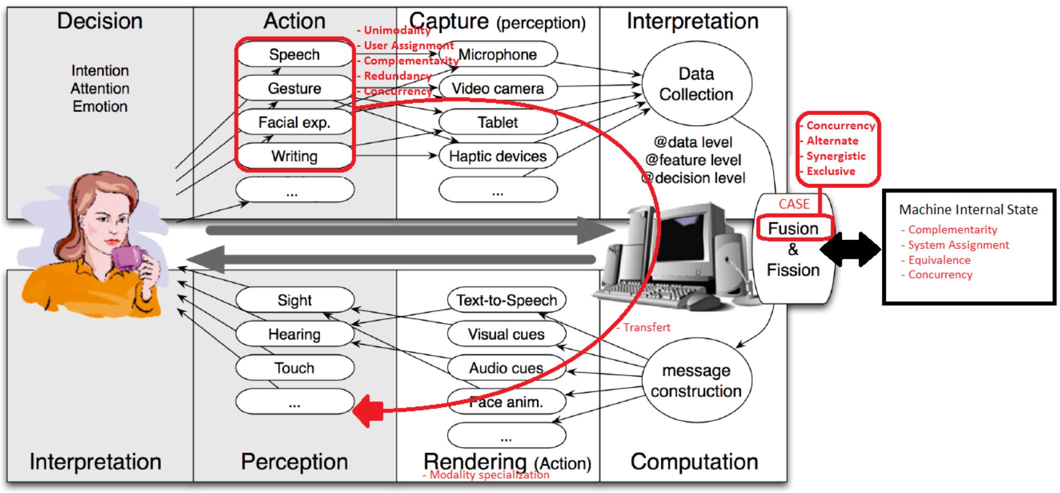


Figure 4.1: Placement of the properties

the moment where the rendering is generated.
The Transfer is a big arrow, one modality is interpreted, computed and then returned to the user in another modality.
The CASE model is at the fusion engine level because it is all the properties it should have to be able of handling the various forms of the user input and to give it to the system (state machine).
Unimodality, User Assignment, Complementarity, Redundancy and Concurrency are near the user, because they explain how the user can use the system.
Complementarity, system assignment, equivalence and concurrency explain how the system can change its state.

4.4 Going to a new property organization

Now from the Figure 4.1 we can go to the user properties explaining all the different ways the user can use a multimodal system and all the functions a multimodal system can offer at a defined time, with the logical organization between those. This organization is important because a system should offer multiple ways to use it, but the user will often choose only one of them. We have established three dimensions for the analysis of our properties : the number of modalities, the temporal relationship and the semantic combination between those modalities.

4.4.1 Multimodal command unity

A multimodal command unity is the time allowed by the system for one command before going to another. It is very important for asynchronous use of modalities. It makes the difference between the redundancy and the execution of two times the same command. We present the properties for the use of the system by the user at a time T and the possibilities offered by the system at the same time for a given multimodal command unity. As presented in the MUDRA article [16], the architecture of multimodal systems can be Data Stream-Oriented or Semantic Inference-Based [16]. The first architecture deals with the fusion of data flows when the second deals with semantic fusion (with state machines among other things). Our multimodal command unity is more useful with the second one.

4.4.2 User's decision tree

We will now examine the user's interaction possibilities. First we make a distinction between the type of communication. It can be unimodal or multimodal but only one at the same time. Then if the communication is multimodal we make a distinction on the temporal relationship between the use of the modalities: it can be synchronous or asynchronous. The synchronous use can be complementary, redundant or concurrent, while the asynchronous one can only be complementary or redundant. The Figure 4.2 shows a graphical representation of that organization.

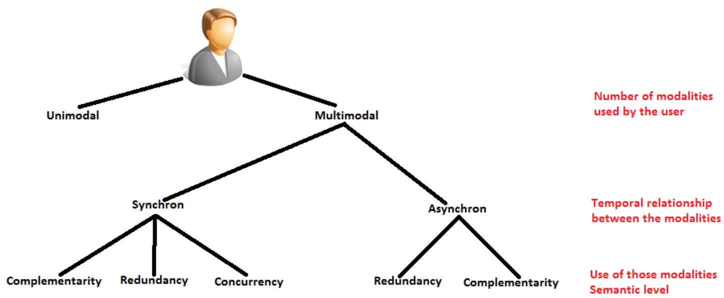


Figure 4.2: Human to system communication

4.4.3 System's decision tree

On the machine side, we will focus on what the system can offer to the user. First, like for the user, we make a distinction between the type of communication: it can be unimodal or multimodal but only one at the same time (beware, a system can offer to handle multimodal and unimodal commands at the same time, in that case we have a multimodal offer). Then, if the communication is multimodal, we differentiate the potential temporal relationships that should be handled by the system (synchronous or asynchronous). For the asynchronous property the system may impose an order for the commands. After on the semantic level, the only case where the order matters is when the asynchronous modalities are used in a complementary way. For the Equivalence itself, since only one modality should be used at a time, the synchronous/asynchronous has no importance, but it needs to be related to since Redundancy results from Equivalence and can be synchronous or asynchronous. The synchronous use can also correspond to a concurrent or complementary use of the modalities. The Figure 4.3 shows a graphical representation of that organization.

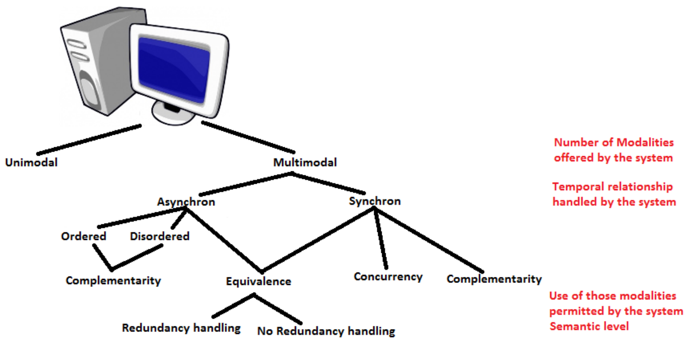


Figure 4.3: System possibilities

4.4.4 Unified schema

Now that we have the two sides of communication, we can unify them showing the relations between each other. When the user uses a unimodal way of communication, this is not always because the system only allows it. The system may offer multiple ways of entering the commands (Equivalence with Redundancy handling or not), and the user chooses one of them. The synchronous and asynchronous Redundancy and the Concurrency naturally plug in each other. For the asynchronous Complementarity, the user has no notion of ordered command unless it is imposed by the system. The order has no particular meaning for him. So, this notion is not on the user's side, and the user asynchronous Complementarity corresponds to the system asynchronous ordered or disordered Complementarity. The synchronous Complementarity naturally plugs on each other.

Since the mapping is a bit chaotic in Figure 4.4, the Figure 4.5 shows another organization with the same properties. We just switched the temporal and semantic dimensions, and the mapping looks more understandable. The other advantage is that having the synchronous and asynchronous aspects in the last place permits to differentiate the Data Stream-Oriented or Semantic Inference-Based architecture [16].

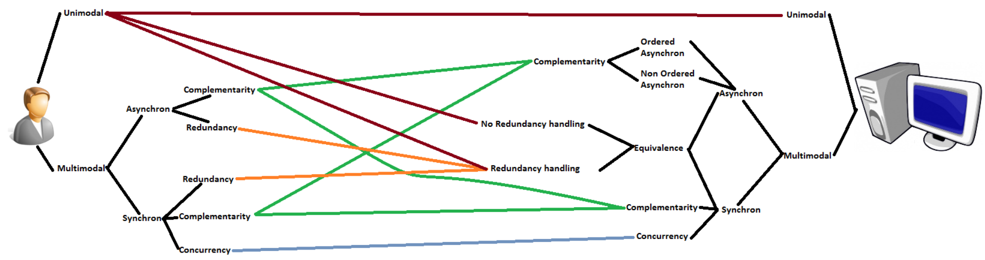


Figure 4.4: Combined Schema

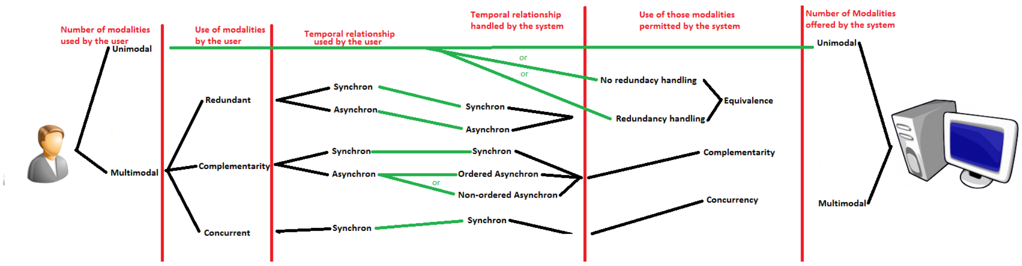


Figure 4.5: Combined Schema

4.4.5 Positioning compared to the other models

We propose a unified model, something that sorts in a clear way the different aspects of the previous models and differentiate the user and system properties. We also wanted a model where all the interactions possibilities are represented. We still find the majority of the properties from the CASE, CARE and TYCOON models in our model.

CASE

The Concurrent property is there on the computer and the user side.
The Alternate property is the Complementary asynchronous property from the user and system side. Our model adds the notion of order to the machine side of that property.
The Synergistic property is the Complementary synchronous property from the user and system side.
The Exclusive property goes in the unimodal use of the user and the equivalence and unimodal possibility from the system side.

CARE

The Complementarity property stands on both sides. We have combined it with the Alternate and Synergistic property from the CASE in our model to eliminate the overlapping between the CARE and CASE.

For the Assignment property, it is a bit more complex, the System Assignment (or Strict Assignment) is on the system side, when the system only offers Unimodality. The user Assignment is when the user uses Unimodality and the system offers Equivalence (with or without Redundancy handling). Since our model describe the use of the system and the possibilities offered at a time T the notion of "the user always uses the same modality" [28] cannot be introduced.
The Redundancy property stands on both sides, we add the fact that the system may not be able to handle Redundancy in every state and the fact that the Redundancy can be synchronous or asynchronous (for example: on a multimodal music player when you press the button play next then say the vocal command "Play next" the system can be able to handle asynchronous redundancy in a given temporal window).
The equivalence property is on the machine side, the system offers several possibilities to the user, who chooses which one(s) he will use.

TYCOON

As detailed in the previous point the Concurrency, Complementarity, Equivalence and Redundancy are in our model.
Since the Transfer property is the least used and implies a primary computation, the Transfer property does not fit in our model. Nevertheless we will speak about this property when we will detail the error recovery.
For the Specialization, the Data Relative Specialization being the same as the System Assignment, it is on our model (see the point over the assignment). On the other hand, the Modality Relative Specialization is not in our model. The Specialization of a modality in some event(s) is a developer choice and cannot be included in our model. Since we do not have modality relative Specialization, we cannot have Absolute Specialization.

4.4.6 Error handling

Our model takes the happy case of multimodality, but in practice, a system should be able to handle errors. When the first multimodal command is not recognized, the command can be entered again, so we have a redundancy between two commands, in two Multimodal command unities.
If a part of the first command is not recognized, the system can require the missing part with the same or another modality, so we have Redundacy or Complementarity between two commands in two Multimodal command unities. This way of handling errors is like the transfer property.

4.5 MUSE

Now that we have our model, we have to give it a name. The model is talking about the user and system properties and about the relations between them. Therefore after some reflexions, we decide to name our model MUSE, for Multimodal User-System Equivalence. This name is really meaningful, in the Greek mythology, the Muses are the daughters of Zeus and Mnemosyne (the memory personified). They are the goddesses of the inspiration of literature, of science and of the arts. This really fit what we want to do with our model, to provide developers and users inspiration of how the modalities can interact.

Chapter 5

Applications and model improvement

5.1 First application - property visualization

With this first version of our model, we create an application giving an example of all the user properties available, even the most complex ones. After some reflexion, we decided to implement a drawing application like the Bolt's Put-that-there [2]. For the technology, we use the Google Javascript API for voice recognition [11] and the paper.js framework for the drawing part of the application [18].
The application consists in a grid where the user can draw red or blue circles or squares using buttons, voice or a combination of both. Since the application has an educational goal, when the user enters a command, the command is placed on the model, with the decomposition of the input mode recognized. All the properties of the model can be illustrated, even combinations. The application is represented on the fig 5.1.
Some examples are :

- Voice : "Draw a circle here" + mouse on a position (synchronous Complementarity)
- Voice : "Draw a circle in 1.2" (Unimodal)
- Voice : "Draw in 4.6" + circle and red selected on the buttons (asynchronous Complementarity)
- Voice : "Draw a circle here" + red selected + mouse on a position (synchronous and asynchronous Complementarity)

This application does not really satisfy us, if we could illustrate all the existing possibilities, it was not really intuitive and the multimodality seemed a bit "forced" (entering the command using only one modality was often

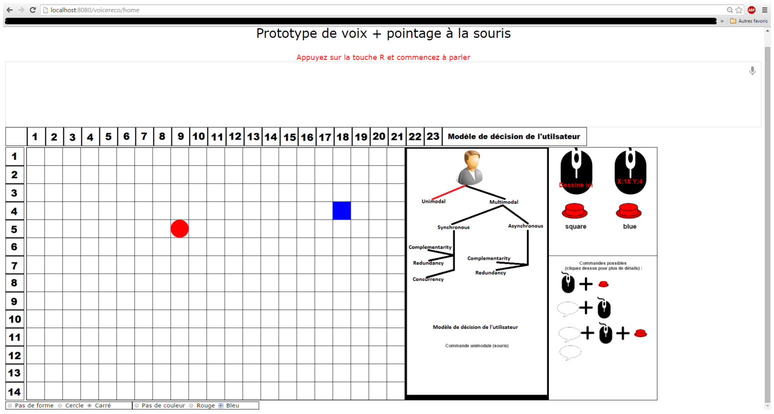


Figure 5.1: First application

easier).

5.2 Second application - Intuitive multimodality

We decided, on the opposite, to make an application where the multimodality is intuitive and funny. To do so, we decided to make a game. The player controls the head of a snake of color with his mouse and can change the width with his voice using the keywords "Small" and "Large". His goal is to fill with color the gap between the lines while being the most precise possible. The application is represented in the fig 5.2.

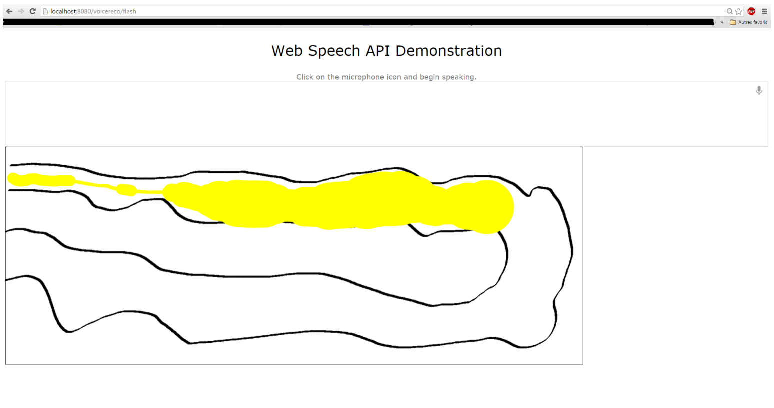


Figure 5.2: Second application

This application is a great reflexion tool. How does the user perceive the task he is performing in this system? Does he use Complementarity, using the two modalities to achieve the goal of filling the white space of color ? The answer is yes but when you look at the system implementation the function of moving the head of the snake and changing its width are two different functions. So even if the user is using the two modalities to achieve a simple goal, like a painter modifying the pressure on his brush, he is using different functions of the system. So this brings an essential point to the model : the fact that the representation of the system of a task and the user's one are not necessary the same.

5.3 Model evolution

After analysis of the two multimodal applications, two elements can be highlighted :

- In front of an application, the user will always interact in the way which is the most intuitive for him. He does not care about the synchronous or asynchronous temporality, the potential handling of redundancy and the orders of the commands; all those constraints come from the system.
- On the opposite, the system will often offer multiple ways to perform a task. That is why the model needs to take all the possibilities in account.

Therefore, we modified the user-side of the model to match the simplicity constraint described above. The new user-side is presented in the fig 5.3.

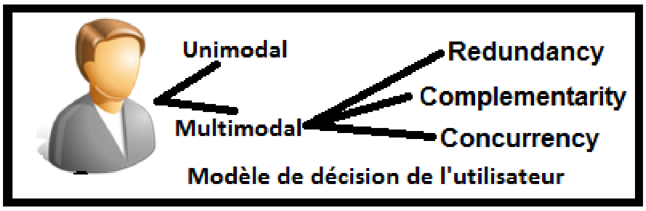


Figure 5.3: Second version of the user side

5.4 Third application

The first application becomes deprecated with the evolution of the model, so we decided to make another one, but with some modifications. It will still be a drawing application but this time we create a multimodal command compositor. The user will have to choose the pieces to assemble in a command and run it. Like for the first application, the commands can be entered using voice or buttons. The goal was the same as in the first application, illustrating all the properties, with a special focus on how the user builds the multimodal command. The application logs all the command to ensure a future analysis. The Figure 5.4 shows a screenshot of the application interface.

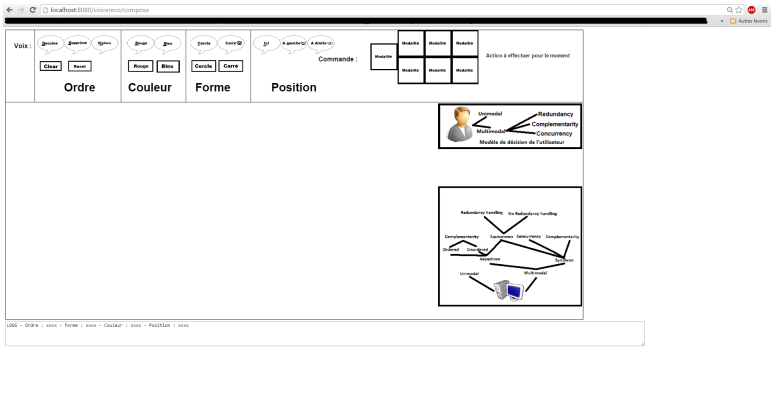


Figure 5.4: Third application

5.5 Conclusion

All those applications fill their role of enhancing the reasoning. We were able to enhance the model by the simplification of non essential aspects, to make a primary evaluation of coherence and completeness. It is really satisfying, we can place all the encountered scenarios on the model with precision, which was our main goal. But there are still some problems, first we need to give it a name, secondly the notion of synchronous and asynchronous still seems a bit complicated and third we still have to evaluate the model with users.

Chapter 6

User evaluation

6.1 Introduction and motivation

The user evaluation is a central point in the development of a new model. It helps to manage the strengths and weaknesses and where the model should be enhanced. In order to do this, we make a survey on the students in Master of Computer Science at the University of Namur, especially the ones who follow the course "advanced methods of interaction" from professor B.Dumas. The evaluation takes place in Switzerland too, in the University of Fribourg, in the department of computer science of the science faculty with the students of professor D. Lalanne.

6.2 Test population

Normally students are not a good population for surveys. But for this special case they fitted the needed test population perfectly. We wanted people that have a good understanding of computer science but are not familiar with multimodality. As said before, there are student in master classes of computer science, aged from 22 to 24. The test was given in the context of the course "advanced methods of interaction" at a particular time, when the professor has explained what multimodality is and what its advantages are but before he explained the main models such as the CASE or CARE. Like this, the students had a global overview of the multimodality and could first understand the different models. This way of making the evaluation let us compare the way the users learn the different models and if our model is clear enough or not.

6.3 Survey elaborating

First, we have to choose the referring models we will use. Since MUSE takes a lot of its inspiration from the CASE and CARE models we will use them. We will skip the TYCOON because it has less impact in the making of MUSE, and taking three referent models will make the evaluation a lot heavier. The idea of the evaluation is to have use cases of multimodal systems and ask students to place them on the different models. It will show the understanding of the CASE, CARE and MUSE and maybe can we apprehend some difficulties of the different models? At the end of the evaluation we ask some feedback from the students on three different aspects : the complexity (from too large to balanced), coherence (from weak to excellent) and completeness (from incomplete to complete) of the models, on a rating from 1 to 5. We also ask if they have some comments, observations, suggestions to provide for the models.

6.4 Systems selection

After elaborating the way our survey will be made, we have to choose the relevant systems we will use for it. The obvious first pick is Bolt's put-that-there system described in his paper [2] and presented in a the section 2.3 of this text. In the same way, the two following obvious picks are the NoteBook system from the CASE paper [28] and the MATIS system from the CARE paper [9]. For the evaluation to be relevant, we still have to pick one or two systems, therefore it can be interesting to have one of the applications developed for MUSE and one independent system. For the first one we will take the second application : the game system. It will be interesting to see if the users differentiate the system and user's properties like we have done or not. As last system, we will take the RASA military one because it is well documented, and it is a system designed for a real use case, with concrete application. We now have five systems, the last thing we have to do before writing the survey is to define the different use cases.

6.5 Use cases selection

6.5.1 Bolt's Put-That-There

We start with R. Bolt's Put-That-There system [2], as you can guess, the first example we choose is the put-that-there itself, then another example of Complementarity and an example of Redundancy. As a result we have :

- The user point the form and utter "Put that" then point a location and utter "there";
- "Move this to the right of the purple square" + pointing the blue;

- "Delete this purple square".

6.5.2 NoteBook

The second system is the Notebook system presented in the CASE paper [28]. Like for the first system, we start with a basic use case, such as a Unimodal command. Next, since the CASE does not handle the Redundancy, we take one or two examples of redundant interaction. The last example is one of Complementarity but where the modalities are sequential to make the emphasis on the alternate property from the CASE model. Consequently, here are the use cases :

- "Clear all";
- "Insert a note here, between the second and the third";
- "Next note" + clicking on the next note button;
- Clicking on the insert a note button, then uttering "between the second and the third".

6.5.3 MATIS

The third system is MATIS (Multimodal Airline Travel Information System) presented in the CARE paper [9]. As usual, we start with a basic use case like the Complementarity, then we follow with another example of Complementarity but with the second modality delayed and lastly we use an example of Concurrency since the CARE does not handle it. In conclusion, here are the use cases:

- "Show me the USAir flights from Boston to this city" along with the selection of "Pittsburgh" with the mouse on the screen;
- clicking on the Denver airport to see the flights, then typing in the dedicated text window "to Boston";
- "Show me the flights from Boston to Pittsburgh" and clicking on the Denver airport to see the flights.

6.5.4 RASA military system

The fourth system is the RASA military system presented in the section 1.7 of this work, we will use the same use cases as the ones described in that section. As a reminder, the use cases are :

- Drawing an arrow starting near the center of the Post-it note the user says: "Romeo-one-bravo is moving in this direction at 20 kilometers per hour." RASA projects this new fact onto the paper map in the form of an arrow labeled "20 kph.";

- Two soldiers in front of the system, one entering command as in the example, the second drawing circles and labeling them (with labels such as mines, no-go area, etc).

6.5.5 MUSE Game system

The last system is the game system presented in the second application, since the application exists for only one purpose there will only be one use case. As a reminder, here is the description of the system : the user moves a with a color pencil. The goal is to fill of color the space between the two lines.
The first modality is the mouse, it determines the position and movements of the pencil.
The second modality is the voice, the user can change the diameter of the pencil using the keywords.

- The user is playing the game trying to fill the maximum space using the two modalities.

6.6 Evaluation in practice

The evaluation took place on the 23rd of March in Belgium and the 15th of April in Switzerland. It was made in three steps : first the students have a description of the three different model and some time to read and assimilate it. Second we give them the description of the systems with an example to let them apprehend the functionalism. And third we give them the use cases to place on the models. The students took approximately one hour to make it. An example of this evaluation can be found in the annexe B.

Chapter 7

User evaluation : correction and analysis

7.1 Use cases analysis and correction

In this chapter we make an analysis on every use case, establishing the temporality of its modalities and its placement on the different models, then we compare our point of view with what the students have answered.

7.1.1 Bolt's system

First, we analyze the answers of the use cases of Bolt's Put-that-there system. The system uses two input modalities : Voice and Pointing. The first use case is

- The user points the form and utters "Put that" then points a location and utters "There";

This is the classical example of complementary multimodality. The Figure 7.1 shows the modalities. In this particular case, it is difficult to place it on

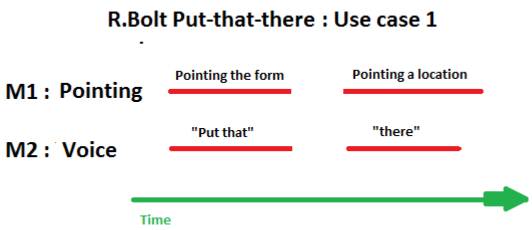


Figure 7.1: "PUT-THAT-THERE"

the CASE model because the user makes a short pause between the "Put

that" and the "There". If the voice flow was continuous we can say it is synergistic but like this it does not fit in the property as represented in the first chapter of this work. It is like a composition of two Synergistic commands. This little pause has fooled many students into the Alternate property, but since the two modalities are working in the same time it cannot be Alternate. This example show again a bit of the limitation of the CASE.
This example is really easy to place on the CARE model: it is Complementarity.
For MUSE now, on the user side it is Complementarity and on the system side it is synchronous Complementarity.
Except for the CASE nearly all of the students correctly placed this example on the models.

The second example for this system is :

- "Move this to the right of the purple square" + pointing the blue

It is a really classical example of Complementarity all the students got the correct answer, the modalities work in the same time (figure 7.2), so it is synergistic for the CASE, Complementarity for the CARE and the user side of MUSE and Synchronous Complementarity for the system side.

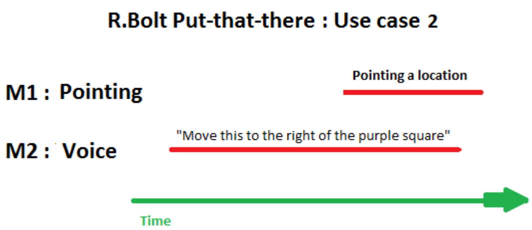


Figure 7.2: Bolt's system, use case 2

The last example for Bolt's system is :

- "Delete this purple square"

In this example the fact that the user was pointing something was implicit by the keyword "this". As a result, many student were fooled into thinking that the command was Unimodal like "Delete the purple square". Again, it is a great example of Complementarity, who puts the emphasis on particular object. The modalities are working at the same time, so it is synergistic for the CASE, it is Complementarity for the CARE and the user side of MUSE. Now on the MUSE's system side, we can have a little discussion : for a computer the command "Delete this" + pointing was enough, the rest is noise or extra-information even if it makes sense for a human. Therefore

Table 7.1: Answers to Bolt's system use cases.

Bolt's System	CASE	CARE	MUSE (User)	MUSE (System)
Use Case 1	Synergistic	Complementarity	Complementarity	Synch. Compl.
Use Case 2	Synergistic	Complementarity	Complementarity	Synch. Compl.
Use Case 3	Synergistic	Complementarity	Complementarity	Synch. Compl.

since the rest of the sentence is used to describe the object that was pointed before it can be seen as redundancy if there is only one purple square. On the other side, the chunk of extra information is not enough to designate the appropriate object if there were multiple ones. So the fact that the command is Redundant or Complementary is determined by the system state at the time the command is given. The Figure 7.3 shows the modalities.

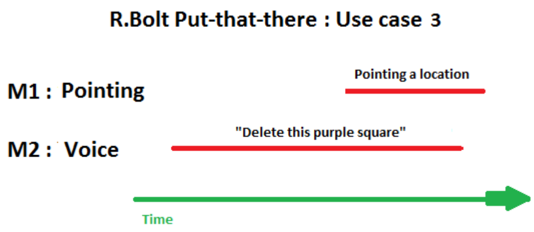


Figure 7.3: Bolt's system, use case 3

The table 7.1 illustrate the correct answers to the Bolt's system uses cases. The color is defined by the proportion of the student who have the correct answer. It is green when the correct proportion is over 75%, orange when it is from 50% to 75% and red when it is under 50%.

7.1.2 NoteBook

The second system on the evaluation was Notebook from the CASE paper [28] and the first use case is :

- "Clear all".

This is really easy, it is Unimodality, the user is only using his voice to clear the NoteBook (shown in Figure 7.4).

One modality, one command, this is exclusive for the CASE, agent assignment for the CARE because the user can use other modalities to do it and simply unimodal for both sides of MUSE. As expected this use case was correctly analyzed by all the students.

The second use case is :

- "Insert a note here, between the second and the third".

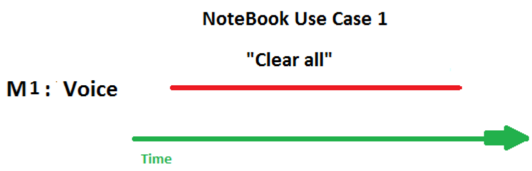


Figure 7.4: Notebook, use case 1

The Complementarity between the keyword here and the pointing is implicit. This example is a bit like the third use case of the Bolt's system. The first half of the sentence with the (implicit) pointing was enough for the system to handle the command, the rest is redundant. In the evaluation the redundancy was detected by less than half of the students. One hypothesis is because the pointing was not explicit even if during the evaluation some students asked if the pointing can be implicit. Another reason is that they did not have a clear mental representation of the modalities. To clarify that look at the Figure 7.5.

Since the CASE does not handle Redundancy even the students who de-

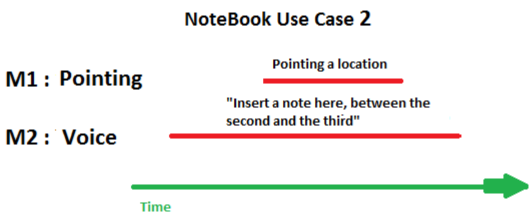


Figure 7.5: Notebook, use case 2

tected it have to find a solution. Surprisingly no one simply passed it. The majority of the students (75 %) matched it with the Exclusive property. For the CARE, the students who detected redundancy indicated it and the others said it was assignment (mostly user assignment). In the same way, for MUSE, people who detected redundancy indicated it in the system side and others said it was Unimodal for both system and user's aspects. One interesting thing is that no one called for redundancy in the user side, even the ones who indicated redundancy in the CARE model (Maybe this is a problem of understanding the model ?).

The Third use case is :

- "Next note" + clicking on the next note button.

Like in the previous example, it was redundancy, but in a clearer way, the 2 modalities are invoked in the same time and have the same meaning (this interaction is graphically represented in the fig 7.6). Again, even if the CASE does not handle redundancy, the students did not skip it and were divided between the Synergistic and Alternate properties. This time the redundancy is indicated by four on five students for the CARE model. For MUSE, unlike the previous example, nearly all the students answered Redundancy in the user model. On the system side of MUSE, surprisingly, students do not go for Redundancy but for Equivalence.

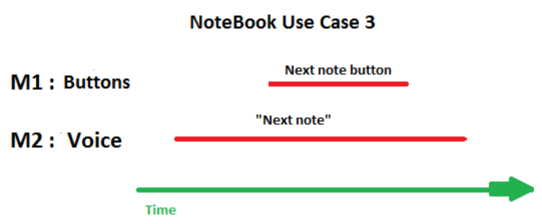


Figure 7.6: Notebook, use case 3

The fourth use case is :

- Clicking the insert a note button, then uttering "between the second and the third"

This example is pretty classic, one modality at a time, but for the same command. It perfectly fit the definition of the Alternate property (CASE model), and nearly all the students find it. It also fit the Complementarity definition for the CARE and both sides of MUSE. The only difficulty was with the synchronous/asynchronous aspect. It seems that it was the most difficult thing about MUSE, since almost half of the students said it was synchronous. It was clearly not, as the "then" in the example indicates a temporal pause and as you can see in the fig 7.7.

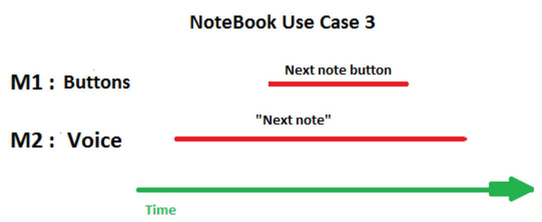


Figure 7.7: Notebook, use case 4

The table 7.2 shows the corrects answers to the Notebook uses cases, the legend for the students answers is the same as for the table 7.1.

Table 7.2: Answers to Notebook's use cases

NoteBook	CASE	CARE	MUSE (User)	MUSE (System)
Use Case 1	Exclusive	Agent Assignment	Unimodal	Unimodal
Use Case 2	KO	Redundancy	Redundancy	Synch. Redundancy
Use Case 3	KO	Redundancy	Redundancy	Synch. Redundancy
Use Case 4	Alternate	Complementarity	Complementarity	Complementarity

7.1.3 MATIS

The third system from the evaluation was MATIS from the CARE paper [9] presented in the previous sections.

The first command is :

- "Show me the USAir flights from Boston to this city" along with the selection of "Pittsburgh" with the mouse on the screen.

There is nothing special here, as shown in the fig 7.8 it is a classical example of Synergistic (CASE) and Complementarity (CARE and both side of MUSE) property. This example is fine for the majority of the students except for the synchronization aspect of MUSE as usual in this evaluation. Similarly, some students mistaken the Synergistic property for the Alternate in the CASE model.

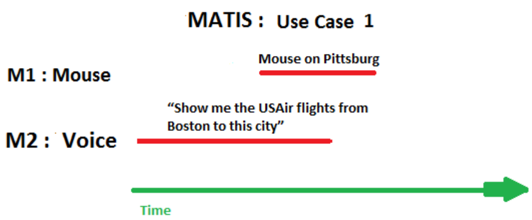


Figure 7.8: MATIS, use case 1

The second command is :

- Clicking on the Denver airport to see the flights, then typing in the dedicated window "to Boston"

This example is similar to the previous one but the modalities does not occur at the same time as represented in the fig 7.9. Students found it quite difficult to find the Alternate property in this example. Only 40 % of them found it, others go for the Exclusive or Synergistic property and some even say that the CASE is not applicable.

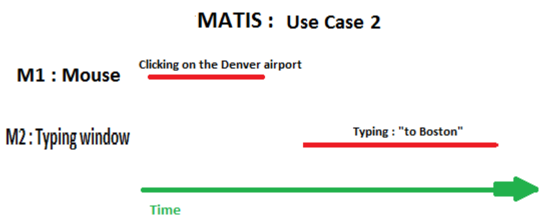


Figure 7.9: MATIS, use case 2

Then for the CARE model the situation is slightly better: more than 60 % of the students find the Complementarity. The rest says it is Agent Assignment.
MUSE is accumulating the problems of the CARE and CASE models. The students who answer Complementarity on the CARE section correctly indicate it in the user side. Again there are a lot of errors with the synchronization aspect. The others say it was Unimodal. On the system side we get plenty of different answers: Concurrency,Unimodality, Equivalence, Redundancy and Complementarity. The correct answer is Complementarity, but after analyzing all the different answers from the students of this use case, it seems that it is very confusing for them. The fact that the two modalities are : first the keyboard then the mouse probably confuse the students because it is the way we classically use our computers and let them think it was only one modality. This explains the fact that we got some "unimodal" and "equivalent" answers. One other problem is that some students miss the temporal pause indicated by the keyword "then" in the sentence.

The third example is :

- "Show me the flights from Boston to Pittsburgh" and clicking on the Denver airport to see the flights.

This is a basic example of concurrency : asking for one thing with the voice while checking another thing with the mouse. As usual, we start with the CASE model. Half of the students find it was Concurrency, others say it was Exclusive and some skipped it. The Exclusive answer is comprehensible if you miss the fact that the two modalities timely overlap themselves. The fig 7.10 shows a graphical representation of that overlapping. For the CARE then, since it does not handle Concurrency, the students are lost. As observed before, only one student simply skipped it, others tried to fit this example in the model, particularly in the Complementarity property. The ones who answered alternate for the CASE where coherent by indicating it is Agent Assignment. For MUSE now nearly all the students finds it is concurrency for both sides of the model, even some who do not found it in

Table 7.3: Answers to MATIS use cases

MATIS	CASE	CARE	MUSE (User)	MUSE (System)
Use Case 1	Synergistic	Agent Assignment	Complementarity	Synch. Complementarity
Use Case 2	Alternate	Complementarity	Complementarity	Asynch. Complementarity
Use Case 3	Concurrency	KO	Concurrency	Concurrency

the CASE model.

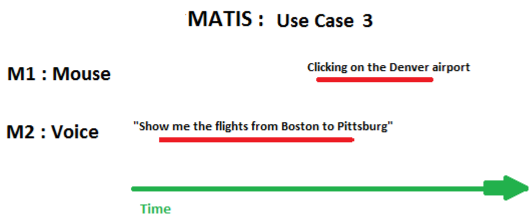


Figure 7.10: MATIS, use case 3

The table 7.3 shows the corrects answers to the MATIS uses cases, the legend for the students answers is the same as for the table 7.1.

7.1.4 RASA

The first use case is :

- Drawing an arrow starting near the center of the Post-it note, the user says : "Romeo-one-bravo is moving in this direction at 20 kilometers per hour". RASA projects this new fact onto the paper map in the form of an arrow labeled "20 kph".

This is a basic example of how Complementarity can be used in an application system. This example does not cause many problems to the students, some of them answer Alternate for the CASE model in place of Synergistic missing the fact that the modalities occur at the same time. For the CARE, all students correctly find the Complementarity. The same goes for MUSE excepted for the synchronous/asynchronous aspects of the modalities. The fig 7.11 shows a graphical representation of those modalities.

The second use case is the last presented and analyzed in the section 3.1.3 which is :

- Two soldiers in front of the system, one entering command as in the example, the second drawing circles and labelling them (with labels such as mines, no-go area, etc.)

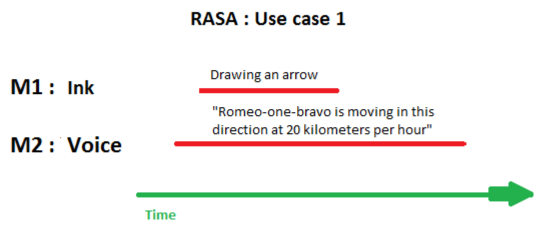


Figure 7.11: RASA, use case 1

Table 7.4: Answers to RASA use cases

RASA	CASE	CARE	MUSE (User)	MUSE (System)
Use Case 1	Synergistic	Complementarity	Complementarity	Synch. Complementarity
Use Case 2	Concurrency	KO	1 : Complementarity 2 : Unimodal	Concurrency

The goal of this example is to see how the students think the model adjusts itself in a multi-user environment. The complete analysis will not be made here again, see the section 1.7.3 of this document. But some elements of how MUSE can handle this type of situation will be added.
Without any surprise, for the CASE model more than 90 % of the students have chosen Concurrency which is the most logical answer when you consider the system globally and you have to match it with only one property (which is how the student interpreted the question).
For the CARE now, half of the student skip it or answers "not applicable". The other half match the use case with Complementarity. The interesting thing is that Complementarity is the evolution of the Synergistic and Alternate properties and no one mention it in the CASE answer.
With MUSE, things become really interesting. On the system side, like the CASE nearly all the students go for Concurrency. Then on the user side, 40 % of the student answer Complementarity. This is not the first time there are different answers between the user and the system sides. But this one is really meaningful. For the students, the users are using the system with Complementarity but as there are 2 users the system perceives it as a Concurrent use. This is really what MUSE is meant for, decoupling the user from the system, even if in this case this is a bit forced by the fact that there are two users. The answer of the remaining 60% are essentially Concurrency, but with some others non-relevant answers.

The table 7.4 shows the corrects answers to the RASA uses cases, the legend for the students answers is the same as for the table 7.1.

Table 7.5: Answers to the MUSE game system

MUSE Game System	**CASE**	**CARE**	**MUSE (User)**	**MUSE (System)**
Use Case 1	Synergistic	Complementarity	Complementarity	Synch. Complementarity Synch. Concurrency

7.1.5 MUSE Game system

The last system is our personal system presented in the chapter five of this document. The use case is :

- "The user is playing the game trying to fill the maximum space using the 2 modalities.

For the CASE and CARE model it is a classical example : students correctly answered Synergistic and Complementarity, the fig 7.12 represent the graphical representation of those modalities. The interesting thing is in the MUSE model. Do some students dissociate the system and user side as we explain it in the third chapter of this document ? The answer is yes for one students out of three. They all answer Complementarity for the user side and for the system side the majority answers Complementarity and the remaining Concurrency.

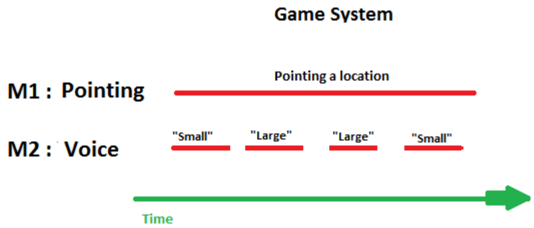


Figure 7.12: Game system

The table 7.5 shows the corrects answers to the MUSE Game System uses cases, the legend for the students answers is the same as for the table 7.1.

7.2 User feedback

At the end of the evaluation we asked the students their feedback on the different models on three different aspect, the complexity, the coherence

Table 7.6: Students feedback

CASE	Ratings :	**1**	**2**	**3**	**4**	**5**
Complexity	Nbr of Students	0	2	3	10	5
Coherence	Nbr of Students	0	1	3	10	6
Completeness	Nbr of Students	0	7	4	9	0
CARE	Ratings :	**1**	**2**	**3**	**4**	**5**
Complexity	Nbr of Students	0	1	8	8	3
Coherence	Nbr of Students	0	3	6	9	2
Completeness	Nbr of Students	0	3	9	4	4
MUSE	Ratings :	**1**	**2**	**3**	**4**	**5**
Complexity	Nbr of Students	4	7	6	1	2
Coherence	Nbr of Students	1	3	7	5	4
Completeness	Nbr of Students	0	1	4	7	8

and the completeness on a Likert scale from 1 to 5. The table 7.6 shows the number of student who gave a determined rating on the models.

7.2.1 Complexity

From this feedback, it appears that the model that is the most "user-friendly" is the CASE with 80 % of the user giving a rate of 4 or 5. The following one is the CARE with 60% with the same rate. MUSE appears to be really complex for the users. 58% of them find it really complex (rate of 1 or 2) and 42 percent find it moderately complex with a rate from 3 to 5.

7.2.2 Coherence

The feedback on the coherence is like the one on complexity. The most coherent model for the student is the CASE one with 80 percent with a rate from 4 to 5 then the CARE with 55 % and MUSE with 50% with the same rates.

7.2.3 Completeness

Contrasting to the two previous point, MUSE gets the best score here with 75 percent of the students finding it complete with ratings from 4 to 5. The CARE and the CASE get the same score, with half of the students finding it complete. A bit more than 1 of three students gave a rate of only 2 for the completeness of the CASE model.

7.3 Conclusion

After analyzing the student evaluation, we will now extract some conclusions for the different models.
First, in general, the examples with Redundancy counted the most errors. This is not detected in half of the answers. The problem might come from the understanding of the systems, the examples or an interference with the fact that the CASE which does not handle Redundancy and the CASE model is the first one presented. A summary table of the answer of all the use cases can be found in the annexe A of this work.

7.3.1 CASE

Even if the CASE is considered the easiest model understood the user, in practice this is not the case. The Alternate and Synergistic properties are often mistaken. It is also the model that counts the most skipped answers.

7.3.2 CARE

The CARE is really well understood by the students, it is the model with the fewest errors. Some students indicate in the comment section that they are disappointed by the fact that the model does not handle Concurrency.

7.3.3 MUSE

MUSE is the model that counts the most errors, principally due to the synchronization aspect of modalities. But if you ignore this aspect, the correctness of the answers is a bit less than the one of the CARE model. This aspect is a heritage of the synergistic/alternate properties of the CASE model, which was not really well understood by the students. Is this still really useful ? This will need to be explored further.
One good thing is that there is no skipped question in the MUSE part of the study. The students were able to analyze all the use cases with MUSE where the CASE and in a smaller way the CARE presented some skipped use cases.
In addition to the answers of the last systems, there were sometimes differences between the system and user's sides of MUSE made by the students which show that some of them think that they can be different.

Chapter 8

Model improvements

8.1 Model improvements following the conclusion of the evaluation

The Synchronization aspect of MUSE causes a lot of trouble to the students during the evaluation. This aspect comes from the Synergistic and Alternate properties. These properties were probably important in 1993 when the CASE model was developed, when you had to determine if the modalities should be processed in the same time or alternatively for the allocation of the limited computing resources. Now we have relatively powerful computers and the allocation of resources is a smaller problem. Additionally, the systems in the 90's where not threaded and not multicores. So now computer are able to do multiple things in the same time with a good time optimization of the processor usage. In the CARE model, these properties are fused into the Complementarity property which is the same property but does not take the temporal relationship between the modalities into account. In the evaluation, this temporal relationship was often misunderstood or simply skipped by the students. This aspect is the root of a lot of discussions with the Professor D. Lalanne of the HUMAN-IST research group from Fribourg in Switzerland. We kept this aspect for the system side because we wanted to be as complete as possible. But the role of the model is to be understandable for the majority of the computer scientists, not only for the experts of multimodal interfaces. So, is this completeness gained with the synchronization aspect relevant ? Or does it cost too much expressiveness ?
To answer this question we have to take a look at the model without the synchronization aspect. The user side does not change, it remains the same.

The system side changes a bit, there is no more duplicate properties between the synchronous and asynchronous sides, making the model clearer. A side effect of this is that the ordered/disordered aspects of the asynchronous side

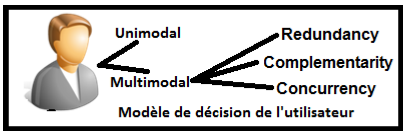


Figure 8.1: Muse, user side

are lost. The reasoning is the same : if we do not care about how the modality input occurs in the time, we should not care about the order they occur. The Figure 4.13 shows how the system side looks.

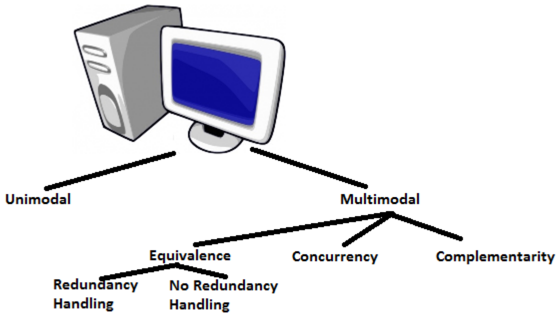


Figure 8.2: Muse, system side

Even without the synchronization aspect the model does not seem to lose any expressiveness, all the use cases of the evaluation can be easily placed again on it which is a thing that some of the students have already done because they have ignored this aspect. Nevertheless, we lose a bit of precision during the transformation. The synchronization and ordering aspects are not high-level enough to be in a clear and understandable model but there are things where multimodal systems designers should pay attention.

8.2 Comparison with the other models

As we said in the Chapter three, the CASE and the CARE models are often understood and presented by the multimodal interfaces course teachers as the system and user sides of multimodality. But this vision does not resist an advanced study. MUSE sorted the different aspects of those models and put the right property in the right place but stays very close to them. There are no new property, just a new arrangement. This analysis is a bit different of the one made after the first version of the model because we give away the synchronization and ordering aspects. This makes MUSE closer to the CARE, and the system side closer to the TYCOON model. We will take a look to the comparison with the other models as we have done with the first version of MUSE. This time, there will be no graph of the matching between the sides of the model, since we observe that the properties do not correspond one to one.

8.3 TYCOON

The TYCOON is kind of a system oriented CARE, with some additional properties. The Equivalence, Redundancy, Concurrency and Complementarity are the same. However the Specialization and Transfer properties are set aside : the Specialization because it is a developer's choice, and the Transfer because it needs multiple states of the system to be observable.

8.4 CASE

This version of MUSE gets a bit away from the CASE since like in the CARE the Synergistic and Alternate properties were fused in the Complementarity and that we have no more distinctions of the temporal organization of the modalities. The Exclusive property goes in the Unimodal one and the Concurrent property goes in both sides of the model. But as we see practically, if Concurrency happens in one side of the model it does not necessarily happen in the second one.

8.5 CARE

The CARE model was divided between the different sides of MUSE : the Equivalence in the system side, the Assignment, Redundancy and Complementarity in both sides. In the Assignment, it is renamed to Unimodal communication.

Chapter 9

Conclusion

The CASE and CARE models are the most used by the multimodal researchers and developers. However, we have demonstrated in this work that some properties are missing and some others are not on the right place. Which explain we have seen in the system section that the developers were not following the models but were making their own appreciation and understanding of it for their needs.
What we have accomplished with MUSE is proposing an alternative to the classical models, sorting the properties to a place that better suits them. The goal was to give developers a map of what it is possible to do in multimodality. We also differentiate the user from the system, showing that the user does not always interact in the same way that the system understands it. This differentiation helps to model multi-users systems too, since the development of new technologies which require this differentiation becomes more and more present. The evaluation done in this work shows that we have reached our first goal which is to have a model that helps us to model all the different types of interactions and use cases. This evaluation was made on 20 people, with is a small number but helped us a lot to get some first results on the understanding and possibilities of MUSE comparing to the other models. The feedback of the users has helped to have an understanding of the complexity of the model to non-specialist users and to cut off some unclear aspects of it in order to gain in simplicity.
In future works, a new survey can be done with the modified model on more people to see if the results go in the same way. Since the changes are minimal, the results should not be substantially different.

In order to see if MUSE is better than the other models, another evaluation can be done with another principle : First taking experienced developer but who do not know multimodality and have never developed multimodal applications. Second making them learn one of the models, one group for the CARE, one for the CASE and one for MUSE. Third, defining a multimodal application with a list of features needed which is communicated to the developers. Four defining an evaluation scale of this application in terms of design, usability, ease of use, efficiency, number of errors, etc which is kept secret until the last step of the evaluation. Then asking each team to develop the determined application. Last, when the applications are done, evaluate it with the scale defined in order to determine which application is the best and what are the advantages of each one. This will take far more time that the evaluation done in this work, but it will show how the different multimodal models can influence the development and quality of the multimodal applications.

Bibliography

[1] Ibrahim Aseel and Johansson Pontus. Multimodal dialogue systems for interactive tvapplications. *(ICMI'02*, 2002.

[2] Richard A. Bolt. Put-that-there": Voice and gesture at the graphics interface. *Proceeding SIGGRAPH '80 Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, 1980.

[3] Dumas Bruno, Lalanne Denis, and Oviatt Sharon. Multimodal interface : A survey of principles, models and frameworks. *WMC02*, 2009.

[4] Mark Cartwright. Muse-definition. `http://www.ancient.eu/muse/`, 2012.

[5] Yu Chen and H. Ballard Dana. A multimodal learning interface for grounding spoken language in sensory perceptions. *ICMI 03*, November 2003.

[6] Schmandt Christopher and Hulteen Eric A. The intelligent voice-interactive interface. *CHI '82 Proceedings of the 1982 Conference on Human Factors in Computing Systems*, 1982.

[7] Schmandt Christopher and Hulteen Eric A. Put that there now: Group dynamics of tabletop interaction under time pressure. *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on*, October 2007.

[8] Chandler Corey D., Lo Gloria, and Sinha Anoop K. Multimodal theater : Extanding low fidelity paper prototyping to multimodal applications. *CHI2002*, April 2002.

[9] Joëlle Coutaz, Laurence Nigay, Daniel Salber, Ann Blandford, Jon May, and Richard M. Young. Four easy pieces for assessing the usability of multimodal interaction : the care properties. *Citeseer*, 1995.

[10] Jöelle Coutaz, , Laurence Nigay, Daniel Salber, and Jean Caelen. The msm framework:a design space for multi-sensori-motor systems. *citeseer*, 1993.

[11] Google Developers. Voice driven web apps: Introduction to the web speech api. https://developers.google.com/web/updates/2013/01/Voice-Driven-Web-Apps-Introduction-to-the-Web-Speech-API, 2013.

[12] Bruno Dumas. *INFOM435 - Méthodes d'interaction avancées*. UNamur, 2015.

[13] Bruno Dumas and Denis Lalanne. Discussion with professor b. dumas and d. lalanne, 25th September 2015.

[14] Vernier F. and Nigay L. A framework for the combinaition and characterization of output modalities. *Springer digital library*, 2000.

[15] Architecture Machine group MIT and Bolt Richard A. Richard a. bolt : Put-that-there demo. https://www.youtube.com/watch?v=0Pr2KIPQOKE.

[16] Lode Hoste, Bruno Dumas, and Beat Signer. Mudra: A unified multimodal interaction framework. November 2011.

[17] Coutaz Joëlle and Nigay Laurence. Multifeature systems : From hci properties to software design. *proceedings for the first international workshop on intelligence and multimodality in multimedia interfaces*, July 1995.

[18] Jonathan Puckey Juerg Lehni. Paper.js presentation and tutorials. http://paperjs.org/.

[19] Rekimoto Jun and Nagao Katashi. The world through the computer: computer augmented interaction with real world environments. *Proceedings of the 8th annual ACM symposium on User interface and software technology*, 1995.

[20] Spano Lucio Davide. Defining care properties through temporal input models. *EGMI2014*, June 2014.

[21] Billinghurst Mark. Put that where? voice and gesture at the graphics interface. *WMC02*, November 1998.

[22] Jean-Claude Martin. Tycoon: Theoretical framework and software tools for multimodal interfaces. *Intelligence and Multimodality in Multimedia interfaces, Citeseer*, 1998.

[23] Jean-Claude Martin. Tycoon: six primitive types of cooperation for observing, evaluating and specifying cooperations. *AAAI Technical Report FS-99-03, www.aaai.org*, 1999.

[24] David R. McGee, Philip R. Cohen, and Wu Lizhong. Something from nothing: Augmenting a paperbased work practice via multimodal interaction. *ACM digital librairy*, April 2000.

[25] Johnston Michael, Bangalore Srinivas, Vasireddy Gunaranjan, Stent Amanda, Ehlen Patrick, Walker Marilyn, Whittaker Steve, and Maloor Preetam. Match: An architecture for multimodal dialogue systems. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, July 2002.

[26] G. R. S. Murthy and R. S. Jadon. A review of vision based hand gestures recognition. *International Journal of Information Technology and Knowledge Management*, July 2009.

[27] Laurence Nigay. Design space for multimodal interaction. *Building the Information Society, Springer*, 2004.

[28] Laurence Nigay and Joëlle Coutaz. A design space for multimodal systems : Concurrent processing and data fusion. *INTERCHI 93*, April 1993.

[29] Sharon Oviatt. Ten myths of multimodal interaction. *Communications of the ACM, ACM digital librairy*, November 1999.

[30] Cohen Philip and Oviatt Sharon. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Communications of the ACM*, March 2000.

[31] Cohen Philip R., Johnson Michael, McGee David, Oviatt Sharon, Pittman Jay, Smith Ira, Chen Liang, and CLow Josh. Quickset : Multimodal interaction for distributed applications. *ACM Multtimedia*, 1997.

[32] Palanque Philippe and Schyn Amélie. A model-based approach for engineering multimodal interactive systems. *INTERACT'03*, 2003.

[33] Patrick Tchankue, Vogts Dieter, and Wesson Janet. Design and evaluation of a multimodal interface for in-car communication systems. *SAICSIT '10,*, October 2010.

[34] Hurtig Topi. A mobile multimodal dialogue system for public transportation navigation evaluated. *MobileHCI'06*, September 2006.

[35] Mukai Toshiro, Seki Susumu, Nakazawa Masayuki, Watanuki Keiko, and Miyoshi Hideo. Multimodal agent interface based on dynamical dialogue model –maico:multimodal agent interface for communication–. *CHI Letters vol 1, 1*, 1999.

[36] Ait Ameur Yasmine and Kamel Nadjet. A generic formal specification of fusion of modalities in a multimodal hci. *Springer digital library*, 2004.

[37] Trabelsi Zouheir, Cha Sung-Hyuk, Desai Darshan, and Tappert Charlers. A voice and ink xml multimodal architecture for mobile e-commerce systems. *WMC02*, September 2002.

List of Figures

List of Tables

Appendix A

Survey Answer table

Table A.1: Answer to the different systems use cases.

Bolt's System	**CASE**	**CARE**	**MUSE (User)**	**MUSE (System)**
Use Case 1	Synergistic	Complementarity	Complementarity	Synch. Compl.
Use Case 2	Synergistic	Complementarity	Complementarity	Synch. Compl.
Use Case 3	Synergistic	Complementarity	Complementarity	Synch. Compl.
NoteBook	**CASE**	**CARE**	**MUSE (User)**	**MUSE (System)**
Use Case 1	Exclusive	Agent Assignment	Unimodal	Unimodal
Use Case 2	KO	Redundancy	Redundancy	Synch. Redundancy
Use Case 3	KO	Redundancy	Redundancy	Synch. Redundancy
Use Case 4	Alternate	Complementarity	Complementarity	Async. Complementarity
MATIS	**CASE**	**CARE**	**MUSE (User)**	**MUSE (System)**
Use Case 1	Synergistic	Agent Assignment	Complementarity	Synch. Complementarity
Use Case 2	Alternate	Complementarity	Complementarity	Asynch. Complementarity
Use Case 3	Concurrency	KO	Concurrency	Concurrency
RASA	**CASE**	**CARE**	**MUSE (User)**	**MUSE (System)**
Use Case 1	Synergistic	Complementarity	Complementarity	Synch. Complementarity
Use Case 2	Concurrency	KO	1 : Complementarity 2 : Unimodal	Concurrency
MUSE Game System	**CASE**	**CARE**	**MUSE (User)**	**MUSE (System)**
Use Case 1	Synergistic	Complementarity	Complementarity	Synch. Complementarity Synch Concurrency

Appendix B

Survey

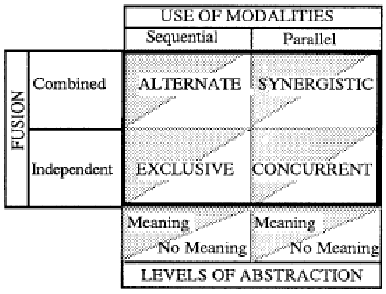
The next six pages are the survey over the multimodal models and systems described in this work.

Multimodal Models Presentation

What is a multimodal system ?

A Modality refers to the type of communication channel used to convey or acquire information. It also covers the way an idea is expressed or perceived, or the manner an action is performed. A multimodal system is a system who uses multiple modalities as input (Definition from the article "A Design Space For Multimodal Systems: Concurrent Processing and Data Fusion" from L. Nigay).

1. CASE – L. Nigay (1993).



Multiple modalities can be used in 4 modes.

Concurrency : the modalities are used simultaneously but with different commands.

Alternate : the modalities are used 1 by one but for the same command.

Synergistic : the modalities are used simultaneously in the same command.

Exclusive : the modalities are used 1 by one with different commands.

2. CARE – L. Nigay, J. Coutaz (1995).

Like the CASE, the modalities can be used in 4 different modes :

Complementarity : The command is entered in the system using a combination of multiple modalities.

Assignment : One modality is used to enter the command, there are 2 types of Assignment :

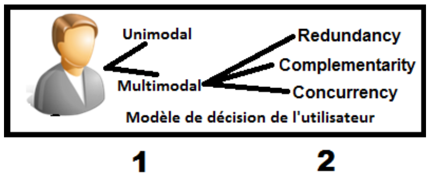
-System assignment : Only one modality can be use to enter the command.

-Agent assignment : Multiple modalities are allowed, but the user always chooses the same.

Redundancy : The same command is entered multiple times simultaneously using different modalities.

Equivalence : The same command can be entered using 2 different modalities.

3. MUSE (Multimodal User-System Equivalence)



The model take into account both the user and system side.

- User side (As simple as possible) :

1. Unimodal communication or Multimodal communication.

2. If Multimodal, the user can interact in

Concurrency (saying something while doing another thing)

Complementarity (using 2 combined modalities to perform a task)

Redundancy (using 2 modalities to do the same thing to be sure that the system performs the right task)

- System side (As complete as possible) :

As the user model, the first distinction is between the unimodal and multimodal communication. Then, in the system's side we have to differentiate the synchronous (using the modalities at the same time) and asynchronous (using one modality and have the other parameters previously checked with another modality) communication.

The asynchronous communication can be :

- Equivalent (some aspects of a request can be pre-checked or given when the request is entered) with or without redundancy handling.

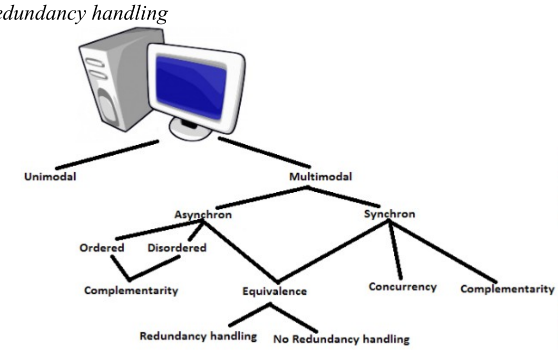
- In the Asynchronous side, an order may be imposed in the modalities, which lead into ordered or disordered complementary (with pre-checked options for example of ordered asynchronous complementarity)

The synchronous communication can be :

- Equivalent (multiple possible modalities to enter one command) with or without redundancy handling

- Concurrent (multiple modalities to enter multiple commands at the same time)

- Complementary (multiple modalities required to enter one command)



Multimodal Systems evaluation

0. Instructions.

5 different systems will be presented in the following section. For those systems, some commands/uses cases will be indicated in the corresponding sheet, you have to place them on the 3 different models presented before, you can use multiple properties.

1. Bolt's Put-that-there system.

Using Bolt's put-that-there system.

Input modalities :

-Voice : command + relative position

-Pointing : absolute position

Example : "Draw a green circle here."

CASE : Synergistic

CARE : Complementarity

MUSE : - User : Multimodal Complementarity

- System : Multimodal Synchronous Complementarity

2. NoteBook.

NoteBook is a personnal electronic book using a continuous multilocutor speech recognition system. It allows a user to create, edit, browse, and delete textual notes. It also contains a few dedicated buttons to create, delete and browse the notes.

Example : "Delete the next note"

CASE : Exclusive

CARE : Agent Assignment

MUSE : - User : Unimodal

- System : Unimodal

3. MATIS (Multimodal Airline Travel Information System).

MATIS allows a user to retrieve information about flight schedules using speech, direct manipulation, keyboard (the user can type sentences in pseudo-natural language in a dedicated text window) and mouse (the screen displays a map with the airports where the user can click to get more information), or a combination of these methods.

Example : "Show me the flights from Boston to Pittsburgh" and clicking on Pittsburgh to see the flights.

CASE : Not applicable

CARE : Redundancy

MUSE : - User : Multimodal Redundancy

- System : Multimodal synchronous equivalence with redundancy

4. RASA military system.

System composed of a smartboard where a map can be attached (the map needs to be localized in the virtual representation of the world), post-its, an ink and speech recognizer. Uses a dedicated set of icons for recognizing miltary units.

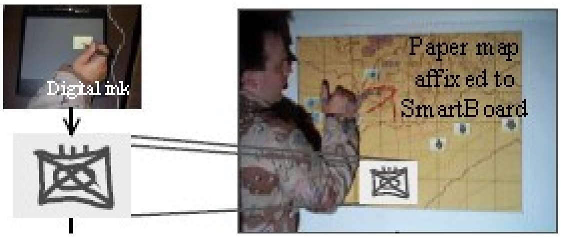
Example : A military draws a unit symbol and at the same time says the unit's name, for example "Romeo-One-Bravo." Next, the user places the Post-it on a registered map of the terrain, in response to which Rasa says, "Confirm: Enemy mechanized regiment called Romeo-one-bravo has been sighted at nine-six, nine-four."

CASE : Not applicable (It's should be a mix between the Synergistic & Alternate properties)

CARE : Complementarity

MUSE : - User : Multimodal Complementarity

- System : Multimodal synchronous & asynchronous complementarity



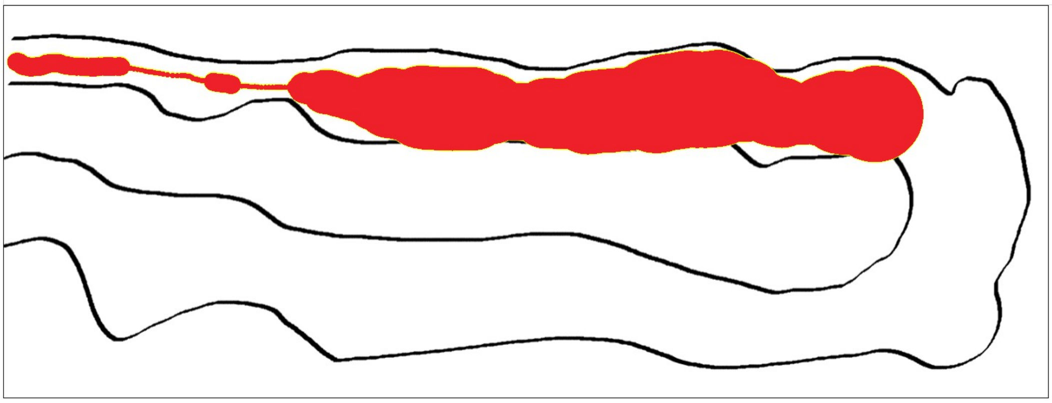
5. Drawing Game.

The user move a color pencil , the goal is to fill with color the space between the 2 lines.

The first modality is the mouse, it determines the position and movements of the pencil.

The second modality is the voice, the user can change the diameter of the pencil using the keywords "Small"/"large"

Use case : the user is playing the game trying to fill the maximum space using the 2 modalities.



Multimodal Systems evaluation

0. Instructions.

Here are the uses cases for the presented systems, you have to place them on the 3 different models presented before as shown in the examples, you can use multiple properties.

1. Bolt's Put-that-there system.

Use Case 1 : The user point the form and utter "Put that" then point a location and utter "there"

CASE :

CARE :

MUSE : - System :

- User :

Use Case 2 : "Move this to the right of the purple square" + pointing the blue triangle

CASE :

CARE :

MUSE : - System :

- User :

Use Case 3 : "Delete this purple square"

CASE :

CARE :

MUSE : - System :

- User :

2. NoteBook.

Command 1 : "Clear all"

CASE :

CARE :

MUSE : - System :

- User :

Command 2 : "Insert a note here, between the second and the third".

CASE :

CARE :

MUSE : - System :

- User :

Command 3 : "Next note" + clicking on the next note button.

CASE :

CARE :

MUSE : - System :

- User :

Command 4 : Clicking the insert a note button, then uttering "between the second and the third".

CASE :

CARE :

MUSE : - System :

- User :

3. MATIS (Multimodal Airline Travel Information System).

Command 1 : "Show me the USAir flights from Boston to this city" along with the selection of "Pittsburgh" with the mouse

CASE :

CARE :

MUSE : - System :

- User :on the screen.

Command 2 : clicking on the Denver airport to see the flights, then typing in the dedicated text window "to Boston"

CASE :

CARE :

MUSE : - System :

- User :

Command 3 : "Show me the flights from Boston to Pittsburgh" and clicking on the Denver airport to see the flights.

CASE :

CARE :

MUSE : - System :

- User :

4. RASA military system.

Command 1 : Drawing an arrow starting near the center of the Post-it note the user says, "Romeo-one-bravo is moving in this direction at 20 kilometers per hour." Rasa projects this new fact onto the paper map in the form of an arrow labeled "20 kph."

CASE :

CARE :

MUSE : - System :

- User :

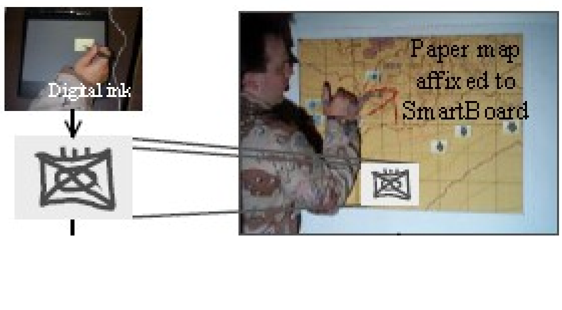
Command 2 : Two militaries in front of the system, one entering command as in the example, the second drawing circles and labelling them (with labels such as mines, no go area, etc).

CASE :

CARE :

MUSE : - System :

- User :



5. Drawing Game.

Use case : the user is playing the game trying to fill the maximum space using the 2 modalities.

CASE :

CARE :

MUSE : - System :

- User :