

Technical Requirements for a Successful Multimodal Interaction

Yacine Bellik
LIMSI-CNRS
B.P. 103
91403 Orsay Cedex France
+33 1 69 85 81 10
bellik@limsi.fr

1 Abstract

Combining several modalities in the same interface requires certain characteristics from input and output devices and the ability to provide some specific information, which is important at the technical level. Unfortunately, several of the current devices do not provide such information. The reason is simple: they have been designed keeping in mind that they will be used in an isolated way, not in combination with other devices. In this paper, we describe the technical requirements revealed by our practical experience when designing multimodal interfaces.

1.1 Keywords

Multimodal Interface, Media Integration, Speech, Gesture.

2 Introduction

Multimodal interfaces are the subject of considerable research worldwide since few years. This paper examines the new problems encountered at the technical level when designing such interfaces. Combining several different modalities in the same interface reveals some new technological problems that are hidden each modality is used in an isolated way. We discuss these problems and describe the requirements for a successful multimodal integration.

3 Multimodal interfaces

Human beings perceive the external world using the five senses of touch, hearing, sight, smell and taste. They interact with it by producing sounds, gestures, etc. In most situations, natural communication between human beings is *multimodal*, as it combines several modes and modalities. Multimodality allows to take benefits in an optimal way of the human communication capacities. *Multimodal interfaces* aim at integrating several communication means in a harmonious way and thus make computer behaviour closer to human communication paradigms, and therefore easier to learn and use. That's why multimodal interfaces are the subject of considerable research worldwide since few years [1]. This has been possible with the advent of multimedia systems that can sample, store and produce complex types of information in real time [2]. There is an ever-increasing effort devoted to multimodal interfaces. The reader can find an extensive description of existing multimodal systems in [3].

4 Importance of the time factor

Timing has a great importance in multimodal interfaces since it can convey information and have effects on the interpretation process of statements. Figure 1 shows that the same interaction process can be interpreted into two different ways depending on the precise temporal distribution of events and in particular on their temporal closeness. The following examples have been encountered in the MEDITOR application (a multimodal text editor) [4]. In the first example, the user points at a first character while saying, "*begin selection*". Then he points at another character while saying "*end selection*". The text between the two characters is then selected. Now, he says, "*bold*". This puts the selected text in bold. Finally, he points at another character and says, "*delete*". Only this last character is then deleted. In the second example, the third pointing operation is done just after saying, "*bold*". This temporal proximity allows the user to specify that the bold attribute must be applied only to the last character indicated and not to the current selection (which is still valid). Since the word "*delete*" is produced alone, it is applied to the current selection.

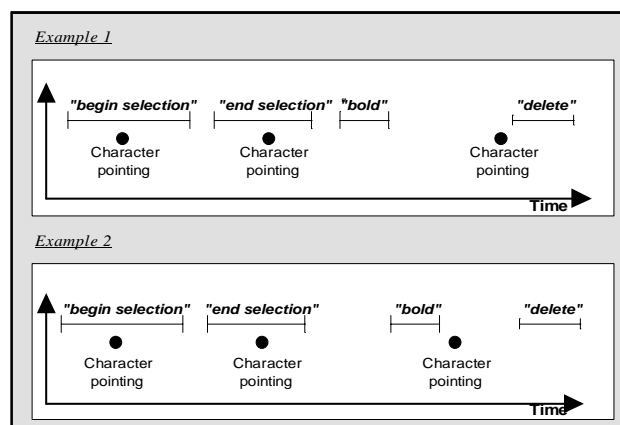


Figure 1: Importance of time factor (1).

Finally, in the first example the selected text becomes bold and the third pointed character is deleted. In the second example we obtain the inverse result though the event

sequences are exactly the same in both cases. We can see in this example that even if the temporal sequence of events is the same in both cases, though their interpretations are different depending on their precise temporal distribution and their temporal proximity.

Let us consider another example which can be encountered in the context of a chemical factory. In the first example of fig. 2, the operator asks the system for the pressure value by pronouncing the word "pressure". The pressure value is then sent through the speech synthesiser. Then he decides to increase the temperature value by pointing at the temperature icon on his touch screen while saying "plus two". In the second example, the operator starts by pronouncing the sentence "pressure plus two". This leads the pressure to be increased by 2 units. Then he points at the temperature icon. The temperature value is then sent through the speech synthesiser.

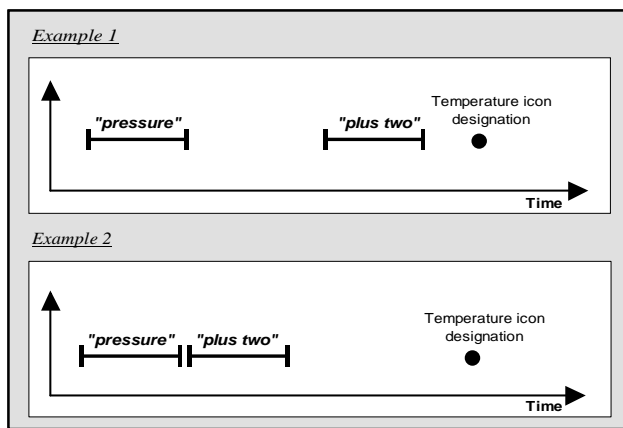


Figure 2: Importance of time factor (2).

So the sequence alone does not allow multimodal statements to be correctly interpreted. It is necessary to have precise information about the temporal distribution of events, in particular the beginning and end times of each word uttered, so it will become possible to detect temporal proximity between them.

5 Temporal Proximity

Temporal proximity between information may indicate a high probability of co-references, which means that data coming from different devices must be merged. It increases the power of the language by adding a new degree of freedom in the expression space. This notion is used since long time in graphical interfaces [5] through the double-click. Double-click is significant only if the two clicks which compose it are close in time. In that case, these two clicks will not be interpreted separately but they will be merged to constitute a new event, which will lead to a new interpretation. However, temporal proximity is not easy to be exploited through a single modality. It is more interesting to exploit it with several modalities.

To define precisely the notion of temporal proximity, it is necessary to analyse the different cases of events temporal sequences. Allen [6] proposed 13 cases (fig. 3).

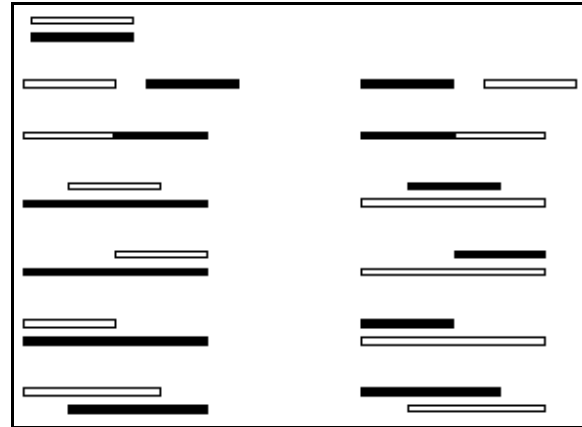


Figure 3: the thirteen Allen temporal relations

These relations are important in the case of multimedia applications [7]. Indeed, they allow to specify precisely the way output information must be synchronised. However, for input multimodality, it is not necessary to distinguish all these cases. The cases described in fig. 4 are sufficient.

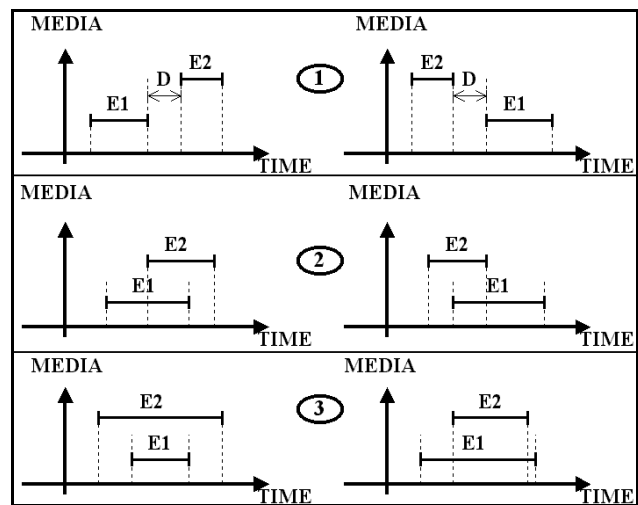


Figure 4: Temporal sequences of 2 events

In the cases 2 (partial covering) and 3 (total covering), it is logical to say that the 2 events are temporally close. In the case 1 (no intersection), the temporal distance between the end time production of the first event and the begin time production of the second event is measured and then compared to a threshold determined experimentally or defined according to user preferences.

It is also possible to use other methods to measure temporal proximity. For instance, let us consider 2 events E and E' and their begin and end times production t_b , t_e , t'_b and t'_e .

The middle distance defined by:

$$Dist(E, E') = ABS\left(\frac{t_b + t_e}{2} - \frac{t'_b + t'_e}{2}\right)$$

consists in representing each temporal interval by its centre and to measure the distances between these centres.

We can also use the Euclidean distance defined by:

$$Dist(E, E') = \sqrt{(t_b - t'_e)^2 + (t'_b - t_e)^2}$$

This is equivalent to represent the events by points in a 2D space. Begin and end times constitute the points coordinates. It is then necessary to try to identify groups of close points (fig. 5).

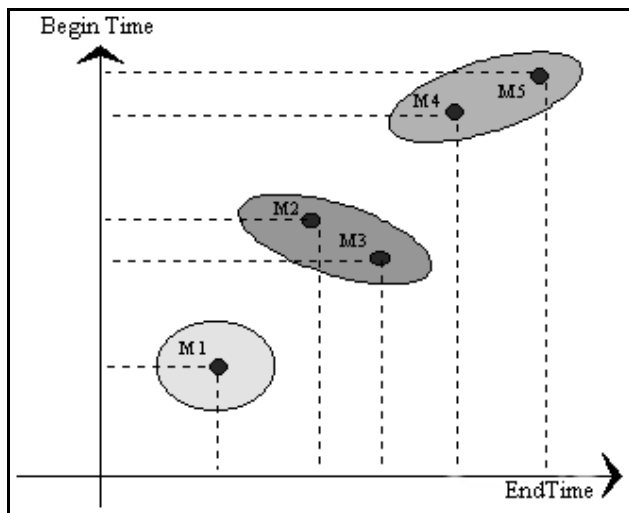


Figure 5: Measurement of temporal proximity using Euclidean distance.

The best method to measure temporal proximity may depend of the used modalities. For instance, it is inadvisable to use the middle distance or the Euclidean distance in the case of 2 modalities such as one of them provides long events and the other provides short events because these distances take into account the duration of events. Thus, the large difference between events lengths may leads to incorrect decisions. A long event and a short event may be considered not close even if one covers the other. However, in the case of the fusion of more than 2 events it may be preferable to use Euclidean distance¹.

¹ In our developments, the method which gave us the best results is the method of the 3 cases described in fig. 4.

6 Response time of devices

In order to interpret the statements produced by the user correctly, it is necessary to handle the events produced by the devices in a sequence, which corresponds to the real chronological sequence² produced by the user. However, the difference between the response times of the different devices can be very important. This implies that, in general, the system receives an information stream in a temporal order which is not correct and which can lead to a bad interpretation of the statements. For instance, if the user says "close" and immediately after this word clicks on a window, then the click event is received before the word event, because the speech recognition system needs more time to recognise a word than the mouse driver does to determine the position of the click event (fig. 2). This could lead to the command being interpreted wrongly.

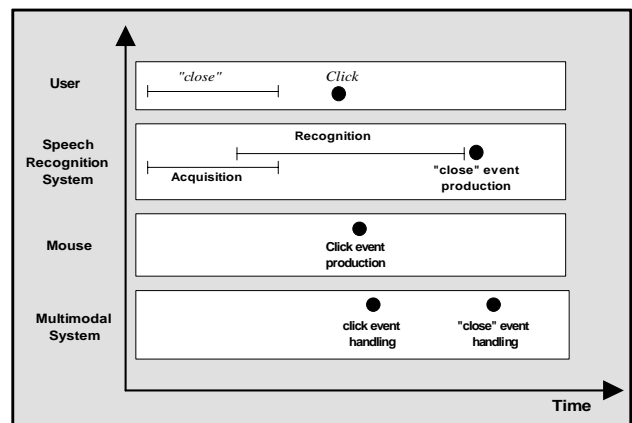


Figure 2: Problem of response time of devices.

To solve this problem, it is first necessary to know the production instants of each word (beginning and end times), so it will be possible to retrieve the right chronological order of events. Then, an event must be handled only after ensuring that no other event is currently being produced by another device, because it is possible that the other event can have an earlier start time. So, we will ensure that the next event produced will have a later time production and we can be sure that events from all devices will be handled in the real chronological order. Concerning speech recognition systems, to check if an event is currently being produced means to check if the user is speaking or if the system is currently doing the recognition process. Unfortunately speech recognition systems do not always provide such information. The same problem is encountered with gesture recognition systems. In both cases it's important to know the user state (is he speaking? is he doing gestures?) and the recognition system state (acquisition, recognition...).

²Even for human beings, it can be hard to understand the meaning of a sentence when the words are mixed up.

7 Early recognition process

Some speech and gesture recognition systems do not start the recognition process until the user's sentence is complete and the acquisition phase is done. We have noticed that it is better that the recognition process starts and delivers results as fast as possible, even if the acquisition phase is still not complete. This allows providing continuous user feedback, which is an important aspect in Human-Computer interfaces.

8 Passive and active co-references

Our practical experience with multimodal interfaces leads us to distinguish two types of co-references:

8.1 Active co-references

Active co-references correspond to the production of two events through two different devices in a sort that the complete and correct interpretation without any ambiguity of one event is impossible without the other. For instance, the user says, "close" while clicking on the title bar of the desired window.

8.2 Passive co-references

Passive co-references correspond to an event production through a particular device in a sort that the complete and correct interpretation without any ambiguity of this event is impossible without knowing the state of another device. For instance, the user says, "close" while looking at the desired window (state of the eye-tracker device).

The problem with passive co-references concerns the necessity to save the state of devices. Let's consider the previous example where the user wants to close a window. Suppose that he can use speech combining with gaze pointing (eye-tracker). To close a window, he can say, "close" while looking at the desired window (fig. 3).

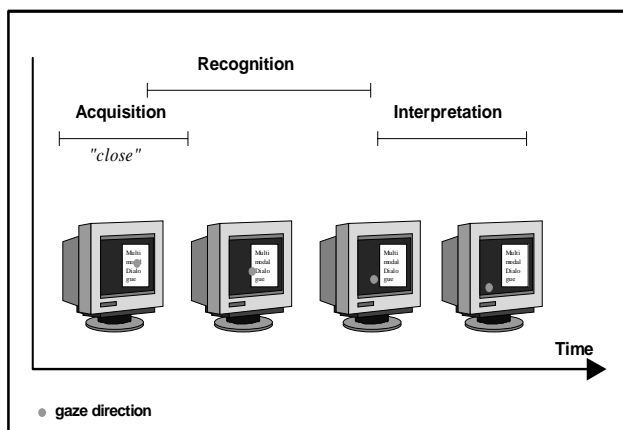


Figure 3: Problem of passive co-references.

If we analyse this interaction from the point of view of the system, we will notice that the speech recognition system needs a certain time to recognise the pronounced word. Even if this time is short, it is possible that the gaze

direction will change before the end of the recognition process. It is then necessary to retrieve the gaze direction when the word has been pronounced else the multimodal system will close another window (the window which corresponds to the current gaze direction). Thus, each device, which has fast state changing, needs to have a large buffer where recent states must be saved.

9 Recommendations to computer device designers

In this section we indicate the main information that computer device designers should provide to allow a real cooperation between modalities and thus a successful multimodal integration:

9.1 Precise event dating

Each event structure must contain a begin time production and an end time production (or a begin time production and a duration). This dating must be precise within $1/10^{\text{th}}$ seconds.

9.2 Device states

It is also necessary to have access to information which indicates the different possible states of a device. For instance, a speech recognition system can be in the following states:

- Waiting: the user is not speaking and the system is not running a recognition process.
- Recording: the user just starts speaking but the system does not yet start recognition.
- Recording and recognising: the user is speaking and the system is running a recognition process.
- Recognising: the user finished speaking but the system is still running the recognition.

These states will allow knowing if an event E is being produced on a given device P. In this case, if another event E' has been produced on another device P' faster than P, then the interpretation of the E' event will be delayed until the recognition of the E event will be over. Indeed, the E event may influence the interpretation of the E' event.

9.3 State history

The devices which have fast state changing (eye-tracker, mouse...) must have a buffer where recent states can be saved. It is recommended to have a buffer which allows to store the device states during 4 seconds with a frequency equal to 50 Hz.

10 Conclusion

This paper has described technical problems that may be encountered when trying to combine several modalities in a unique system. These problems concern mainly the current devices, which have not been designed to be used together in a combined way. We have described the specific requirements needed to a successful integration in a multimodal interface. We think that it is important that future devices will be designed, keeping in mind that they will not be necessarily used in an isolated way, but in harmonious cooperation with other devices.

REFERENCES

1. Cole, R., Mariani, J., Uszkoreit, Zaenen, A., & Zue, V. *Survey of the state of the art in human language technology*. Cambridge, MA Cambridge University Press, 1997.
2. Oviatt, S.L., Cohen, P.R. Multimodal systems that process what comes naturally. *Communications of the ACM*, 43 (3), 2000, 45-53.
3. Oviatt, S., Cohen, P., Wu, L. Z., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J., Ferro, D. Designing the user interface for multimodal speech and pen-based gesture applications: State-of-the-art systems and future research directions. *Human-Computer Interaction*, 15 (4), 2000, 263-322
4. Bellik, Y. Interface Multimodales: Concepts, Modèles et Architectures. *Phd. Thesis*, Paris XI University, France, 1995.
5. Foley, J. D., Van Dam, A., Feiner, S. K., Hughes, J. F. *Computer Graphics, Principles and Practice*. Addison-Wesley Publ., Second Edition, 1990.
6. Allen, J. F., Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, Vol. 26, Num. 11, pp. 832-843, Nov. 1983.
7. Guimarães, N. M. R., Correia, N. M., Carmo, T. A., Programming Time in Multimedia User Interfaces, *UIST'92*, Monterey, California, pp. 125-134, 15-18 Nov. 1992.