FOUR EASY PIECES FOR ASSESSING THE USABILITY OF MULTIMODAL INTERACTION: THE CARE PROPERTIES

Joëlle Coutaz, Laurence Nigay,
Daniel Salber

Laboratoire de Génie Informatique (IMAG)
BP 53, 38041 Grenoble Cedex 9, France
{joelle.coutaz, laurence.nigay, daniel.salber}
@imag.fr

Ann Blandford, Jon May,
Richard M. Young

MRC-APU, 15 Chaucer Road, Cambridge,
UK CB2 2EF
{ann.blandford, jon.may, richard.young}
@mrc-apu.cam.ac.uk

KEYWORDS: Multimodal interaction, complementarity, assignment, redundancy, equivalence, usability.

ABSTRACT: We propose the CARE properties as a simple way of characterising and assessing aspects of multimodal interaction: the Complementarity, Assignment, Redundancy, and Equivalence that may occur between the interaction techniques available in a multimodal user interface. We provide a formal definition of these properties and use the notion of compatibility to show how the system CARE properties interact with user CARE-like properties in the design of a system. The discussion is illustrated with MATIS, a Multimodal Air Travel Information System.

1. INTRODUCTION

In recent years, usability testing has been the subject of growing interest resulting in an explosion of methods and techniques. All are fundamentally based on the identification of properties that the system should satisfy, selection of the methods and tools for supporting the usability goals, and evaluation of the desirable properties. The designer's problem is to decide upon an appropriate set of properties to use. General properties such as consistency, observability, and pre-emptiveness are useful criteria for characterising interactive systems, but they do not cover the specifics of new interaction technology such as multimodal user interfaces.

Multimodal user interfaces support interaction techniques which may be used sequentially or concurrently, and independently or combined synergistically (Nigay & Coutaz, 1993). Novel aspects of interactions must be considered, such as the fusion and fission of information, and the nature of temporal constraints on the interactions. In this paper, we propose the CARE properties as a simple way of characterising and assessing aspects of multimodal interaction: the Complementarity, Assignment, Redundancy, and Equivalence that may occur between the interaction techniques available in a multimodal user interface.

These terms have been used rather loosely in the literature. The next two sections provide formal definitions. The discussion will be illustrated with MATIS, a multimodal interactive system developed in collaboration with colleagues at Carnegie Mellon University. In the final section, we use the notion of compatibility between user preferences and system properties to show how the CARE properties interact with user modelling to predict usability during the design of a system.

2. CONCEPTS: THE CARE PROPERTIES

The formal expression of the CARE properties relies on the notions of state, goal, modality, and temporal relationships.

A *state* is a vector of observables, that is, a set of properties that can be measured at a particular time to characterise a situation. A *goal* is a state that an agent intends to reach. An *agent*, e.g., a user, or the system, or a component of the system, is an entity capable of initiating the performance of actions.

A *modality* is an interaction method that an agent can use to reach a goal. To model the expressive power of a modality m, that is, its capacity to allow an agent to reach state s' from state s in one step, we use the function *Reach(s, m, s')*. A sequence of successive steps (or states) is called an *interaction trajectory*. This generic definition of a modality can be interpreted at different levels of refinement. For example, a modality could be specified in general terms as 'using speech', or more specifically as 'using a microphone'. Both of these interpretations are valid.

A *temporal relationship* characterises the use over time of a set of modalities. The use of these modalities may occur simultaneously or in sequence within a *temporal window* , that is, a time interval. Alternatively, only one modality from a set may be used. Let *Pick(s, m, s')*. be a predicate that expresses the use of *m* among a set of modalities to reach *s'* from *s*.

Modalities of a set M are used simultaneously (or in parallel) if, within a temporal window, they happen to be active at the same time. Let *Active (m, t)* be a predicate to express that modality *m* is being used at some instant *t*. The simultaneous use of modalities of a set M over a finite temporal window tw can be formally defined as:

$Parallel\ (M, tw) \Leftrightarrow (Card\ (M) > 1) \wedge (Duration(tw) \neq \infty) \wedge (\exists t \in tw \cdot \forall m \in M \cdot Active\ (m, t))$

where *Card (M)* is the number of modalities in set M, and *Duration(tw)* is the duration of the time interval *tw*.

Sequential events may have to occur within a temporal window to be interpreted as temporally related. If they occur outside this window, then they may be interpreted differently. Modalities are used sequentially within a temporal window if there is at most one modality active at a time, and if all of the modalities in the set are used within the temporal window:

$Sequential\ (M, tw) \Leftrightarrow (Card\ (M) > 1) \wedge (Duration\ (tw) \neq \infty) \wedge \forall t \in tw \cdot (\forall m, m' \in M \cdot Active(m', t) \Rightarrow m' \neq m)) \wedge (\forall m \in M \cdot \exists t \in tw \cdot Active(m, t))$

Temporal windows for parallelism and sequentiality need not have identical durations. The important point is that they both express a constraint on the pace of the interaction. For example, using multiple modalities simultaneously may be appropriate for the user but may require extra processing resources from the system side or imply a specific software architecture (Nigay & Coutaz, 1993). The absence of temporal constraints is treated by considering the duration of the temporal window as infinite.

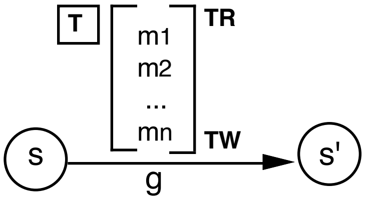

Figure 1: A notation for expressing CARE properties.

Figure 1 shows the relationships between the concepts used in CARE: an agent having reached state s, may have (or has) the goal g to reach state s' by means of the non-empty set of modalities M={m1, m2 ... mn}. These modalities are linked by temporal relationship TR (TR::=|| | ; | 1, for parallelism, sequentiality, and selection of one interaction technique respectively), and constrained by a temporal window TW (TW::=<interval> | ∞). The symbol T at the upper left denotes the CARE property (T::=C | A | R | E).

The CARE properties, which characterise four types of relationships between states and modalities, can be defined and illustrated with MATIS (Nigay, 1994). Our discussion will concentrate on input (i.e., from the user to the system) although the definitions hold for output as well.

MATIS (Multimodal Airline Travel Information System) allows a user to retrieve information about flight schedules using speech, direct manipulation, keyboard and mouse, or a combination of these methods, supporting individual and synergistic use of multiple input modalities. For example, using a single modality, the user can say "show me the USAir flights from Boston to Pittsburgh", can type sentences in pseudo-natural language in a dedicated text window, or can fill in a form using the keyboard and the mouse. When exploiting synergy, the user may also combine speech and gesture as in "show me the USAir flights from Boston to this city" along with the selection of "Pittsburgh" with the mouse on the screen.

3. CARE PROPERTIES: FORMAL DEFINITION

Equivalence: Modalities of set M are *equivalent* for reaching s' from s, if it is necessary and sufficient to use any *one* of the modalities. M is supposed to contain at least two modalities. More formally:

$Equivalence\ (s, M, s') \Leftrightarrow (Card(M) > 1) \wedge (\forall m \in M\ Reach\ (s, m, s'))$

Equivalence expresses the availability of choice between multiple modalities but does not impose any form of temporal constraint on them. Figure 2 shows an example of equivalence between several modalities for 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. Alternatively, they may interact with the Tool window and pick up “Pittsburgh” as a destination from the menu of known cities.

Assignment: Modality m is *assigned* in state s to reach s', if no other modality is used to reach s' from s. In contrast to equivalence, assignment expresses the absence of choice: either there is no choice at all to get from one state to another, or there is a choice but the agent always opts for the same modality to get between these two states. Thus we can define two types of assignment:

$\mathit{StrictAssignment}\ (s, m, s')\ \square\ \mathit{Reach}\ (s, m, s')\ \square\ (\square\, m' \square\, M\ (m' \neq m\ \square\ \neg \mathit{Reach}\ (s, m', s')))$

$\mathit{AgentAssignment}\ (s, m, M, s')\ \square\ (\mathit{Card}(M) > 1)\ \square\ (\square m'\ \square\ M\ \mathit{Reach}\ (s, m', s')\ \square\ (\mathit{Pick}\ (s, m', s')\ \square\ m'=m))$

In the case of an agent assignment, it is interesting to analyse the interaction trajectories to explain and justify its occurrence. In MATIS, window manipulation is performed by direct manipulation only. In particular, speech cannot be used as an alternative. Therefore, the system imposes a strict assignment upon the user for window based tasks. Conversely, a user who always uses speech to specify trip destinations would turn the Equivalence offered by the system into an agent assignment. This issue will be developed further in our section on user modelling.

Equivalence and assignment both measure the choice available at some point in the interaction trajectory. Redundancy and complementarity go one step further by considering the combined use of multiple modalities under temporal constraints.

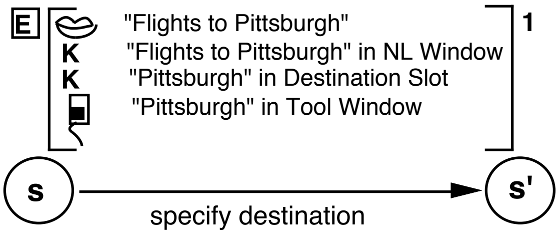
E
"Flights to Pittsburgh"
K "Flights to Pittsburgh" in NL Window
K "Pittsburgh" in Destination Slot
"Pittsburgh" in Tool Window
1
s
specify destination
s'

Figure 2: Examples of equivalence in MATIS. The “lip” symbol denotes speech, K represents the use of the keyboard, and the mouse symbol, a mouse selection.

Redundancy: Modalities of a set M are used *redundantly* to reach state s' from state s, if they have the same expressive power (they are equivalent) and if all of them are used within the same temporal window, tw. In other words, the agent shows repetitive behaviour without increasing its expressive power:

$\mathit{Redundancy}\ (s, M, s', tw)\ \square\ \mathit{Equivalence}\ (s, M, s')\ \square\ (\square m \square M \cdot \square t \square tw \cdot \mathit{Active}(m, t))$

Redundancy can comprise two distinct temporal relationships – sequentiality and parallelism – which may have different implications for usability and software implementation. In particular, parallelism puts restrictions on the types of modalities that can be used simultaneously: modalities that compete for the same system or human resources cannot be activated in parallel. The agent can then only act sequentially if it can comply with the temporal constraints (i.e., it must act quickly for the multiple inputs to be treated as if they were parallel).

For example, the MATIS system is able to support parallel-redundancy between speech acts and any one of the other equivalent modalities presented in Figure 2. As shown in Figure 3, redundant typing in the NL window and in the destination slot competes for the same system (and human) resources. Therefore these two methods must be used sequentially (unless two keyboards and two users were available) and within the temporal window tw, which in turn must be kept within tw'. When the user does not satisfy the temporal constraint, MATIS creates a new request initiated with the extra “redundant” information. This system decision may not be in accordance with the user’s expectation or intention.

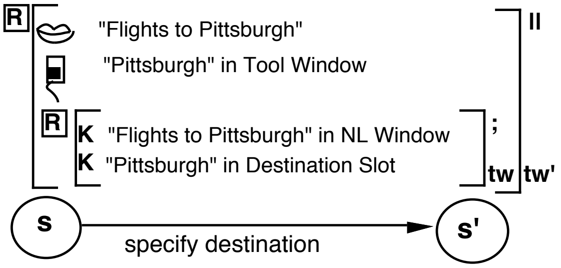
R
"Flights to Pittsburgh"
"Pittsburgh" in Tool Window
R
K "Flights to Pittsburgh" in NL Window
K "Pittsburgh" in Destination Slot
;
tw
II
tw'
s
specify destination
s'

Figure 3: Example of redundancy in MATIS.

According to our formal definition, redundancy requires equivalence but equivalence only stipulates the existence, not the activation, of multiple methods. This asymmetry has implications for the system's robustness. Suppose for example that a user is sending the same content concurrently via the equivalent modalities m and m'. If the system does not support redundancy for m and m', either it is unable to sense m while sensing m', or it senses both of them and may get confused. Therefore, software designers must have a clear architectural model as well as the appropriate software mechanisms to handle these issues appropriately.

Complementarity: Modalities of a set M must be used in a *complementary* way to reach state s' from state s within a temporal window, if all of them must be used to reach s' from s, i.e., none of them taken individually can cover the target state. To express this adequately, we need to extend the notion of reachability to encompass sets of modalities: REACH(s,M,s') means that state s' can be reached from state s using the modalities in set M.

$$
\begin{aligned}
&\textit{Complementarity}\,(s, M, s', tw)\\
&\square\ \ (\textit{Card}(M) > 1)\ \square\ (\textit{Duration}(tw) \neq\ \ )\\
&\square\,(\square M'\square \mathbf{P}M\ (M' \neq M\ \square\ \ \neg \textit{REACH}\,(s, M', s')))\\
&\square\,\textit{REACH}\,(s, M, s')\,\square\,(\square m \square M \cdot \square t \square tw \cdot \textit{Active}(m, t))
\end{aligned}
$$

Deictic expressions, characterised by cross-modality references, are examples of complementarity. As shown in Figure 4, a MATIS user can type or speak the sentence "flights to this city" (or simply "flights to") and select a city name on the screen. Here, the sentence specifies the focus of interest (i.e., the destination of the trip) while the mouse selection denotes a location. These two modalities complement each other and must be combined to reach the intended goal. As with redundancy, complementarity may occur in parallel or sequentially within a temporal window.

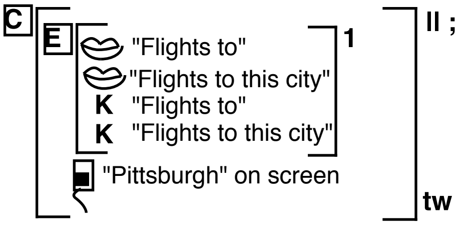

Figure 4: Examples of complementarity in MATIS. Complementarity can be parallel or sequential.

In contrast to redundancy, which does not favour any modality, complementarity may be driven by a dominant modality, which requires the use of others. Typically, in MATIS, deictic references in speech require the use of the mouse to point to a screen object.

Cross modality references may draw upon both complementarity and redundancy. For example, a MATIS user may say "flights to this city" while typing "Pittsburgh" in the destination slot of the request form. In this case, from the system's perspective, the speech act denotes the topic of interest while the typing action specifies both the topic and its value. The speech act, which is covered by the typing act in the destination slot, should be ignored by the system (i.e., the system should not wait for the resolution of the deictic reference). Again, the CARE properties have implications for system implementation.

Another source of complexity for software designers is that distinct actions produced within the same temporal window through different modalities are not necessarily complementary. In this case, fusion must not be performed. For example, a MATIS user may say "Flights to Pittsburgh" while selecting an irrelevant object on the screen. Elsewhere (Nigay & Coutaz 1993; Nigay 1994) we describe how these issues can be supported consistently through a reusable software mechanism.

Our formal definitions of the CARE properties provide conceptual foundations for reasoning about multimodal interaction. To put them to work, we need to discuss issues such as coverage and refinement. Coverage has to do with the set of states for which a particular property holds. For example, in MATIS, the situations described in Figures 2 to 4 hold for any goal related to request specification. Coverage of a property over states can be used as a metric for assessing consistency. The CARE properties, which include the notions of modality and goal, can also be instantiated at multiple levels of refinement. Modality refinement in terms of devices and interaction languages is discussed in (Nigay, 1994).

Typically, goals are recursively decomposed into subgoals. In our modelling technique, this decomposition is expressed as a refinement of interaction trajectories. Depending on the level of refinement, interaction trajectories can be viewed either as a one step encapsulation or as a sequence of steps. For example, in Figure 5 the intended goal is to reach a state where both the departure and the destination of the trip are specified. This goal may be seen as a single chunk or as an encapsulation of two subgoals: specify departure and specify destination.

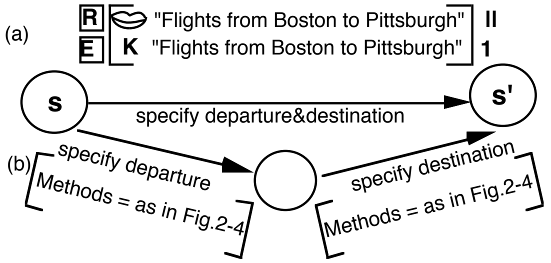
(a)
R
E
"Flights from Boston to Pittsburgh"
K "Flights from Boston to Pittsburgh"
II
1
s
s'
specify departure&destination
(b)
specify departure
Methods = as in Fig.2-4
specify destination
Methods = as in Fig.2-4

Figure 5: CARE and goal refinement.

The user may use one modality, saying the sentence "Flights from Boston to Pittsburgh", or typing it into the speech recognition window, or doing both redundantly (case a). Alternatively, they could specify the departure using any method described in figures 2 to 4 and then specify the destination (case b). In case (a), the user's actions must be performed within some interval and the system provides feedback once the temporal window has elapsed. In (b), the system provides feedback for each subgoal. Considering the description in Figure 5 at a high level, methods (a) and (b) are functionaly equivalent. At a finer grain of analysis however, (a) and (b) differ in the interaction trajectory. As such, they may not be perceived as equivalent.

In summary, the CARE properties can be used at multiple levels of goal refinement. Designers can exploit the recursive nature of CARE to reason about multimodality at the appropriate level. For example, in the early stage of the life cycle, one may reason at a coarse grain to flesh out the most salient requirements about multimodality. Once the system is designed, one may need to go into more details to make sound predictive assessments. Once the system is implemented, effective use of the system can be observed and interaction traces may be analyzed at fairly low level of details. In all of these circumstances, the CARE properties apply.

4. CARE-LIKE PROPERTIES OF THE USER

The CARE properties of the computer system have a counterpart in corresponding properties of the user: the CARE-like properties. As with the system CARE properties, the user properties are concerned with the choice between different modalities for communicating with the computer. As above, our discussion will be confined to the choice of modality for communication in the input direction, that is, from the user to the system.

Because of the user's circumstances – including her task, her background, her training, her knowledge, and the physical and interactive behaviour of the computer interface – the user may well have preferences as to how she communicates with the computer. A familiar example is that if the user is engaged in a task which occupies her hands, she may prefer to communicate with the computer by speech. A little more subtly, suppose that the user wishes to book a flight from somewhere in Europe to Las Vegas. She may not know what is the nearest international airport, so she would prefer to indicate her destination by pointing on a map – or at the very least, by choosing from an appropriately filtered list of airports.

We refer to such preferences by the user, affecting her choice of input modalities, as U-preferences. Certain patterns of U-preferences are worth identifying. (a) If only one modality is acceptable to the user, or if she has a strong preference for one particular modality, then we have a case of U-assignment. (b) If there exists a subset of the possible modalities which she prefers to all others, but between which she is indifferent, then we have a case of U-equivalence. (c) If the user prefers to employ two or more means of communication to convey the same information, then we have a case of U-redundancy. (d) If the user's preference is to use one modality for one aspect of the task and another modality for another aspect, then we have a case of U-complementarity.

The crucial requirement on the design of the system is that its properties must be compatible with the user's U-preferences. In the example above, in order to specify a destination of Las Vegas, a system Assignment to either clicking on a map or providing a filtered list of airports is compatible with the U-preferences, as is a system Equivalence between them. But if the specification of the destination is Assigned to either typing or talking, then the user is in trouble and the system design is incompatible with her needs.

The issue of compatibility can be addressed more systematically by considering the requirements for each of the CARE properties to be compatible with each of the U-CARE properties. We regard a system design as being *compatible* with the user's needs provided there exists at least one modality which is acceptable to both system and user:

(a) *U-assignment*, where the user requires one particular modality Ua. For system Assignment to modality Sa the condition for compatibility is that Sa=Ua, i.e., that the modality assigned by the system is the same as the one the user needs. For system Equivalence or Redundancy over a set Se, the condition is that Ua∈Se, i.e., that the modality the user needs is among those treated as acceptable by the system.

(b) *U-equivalence*, where the user is prepared to use any one of a set of modalities Ue. For system Assignment, the compatibility condition is similarly that the set includes the modality the system is Assigned to, $Sa \in Ue$. For system Equivalence or Redundancy, the condition is that the set Ue shares at least one member with the system Equivalent or Redundant set, $Ue \cap Se \neq Ø$. In this latter case, the user needs to know which modalities the system will find acceptable, i.e., she needs to know the set Se, in order to choose a modality in the intersection. This question of what the user needs to know is discussed below.

(c) *U-Redundancy*, where all modalities in a set Ur are used. The compatibility conditions are similar to those for U-equivalence, provided that the system is prepared to ignore "extra" input. If the system were assigned to a modality Sa, it would presumably be able to ignore input via other modalities, to which it is not currently paying attention. If this is the case, then if $Sa \in Ur$, the properties would be compatible. This is often the case—our workstations do not mind if we talk to them along with our typing or mousing. System Equivalence or Redundancy over the set Se will be compatible with User redundancy provided that at least one of the modalities the user chooses is in the set, i.e., $Ur \cap Se \neq Ø$, again provided that the system can ignore input in modalities not in Se, and that in the case of Equivalence it can ignore inputs beyond one modality in Se.

(d) Complementarity and U-Complementarity (where the user provides part of the information in one modality, and the remainder in one or more other modalities) are best analysed in terms of the properties of the component parts of the message.

These compatibility conditions summarise the main way in which CARE and user CARE-like properties interact in the design of the computer system. They provide a framework which puts the computer and the user on an equal footing, and within which the concrete circumstances of a particular design can be fleshed out to guide the assessment of usability. An important conclusion is that neither the properties of the system alone, nor those of the user alone, determine usability. System modelling can determine the properties of the system, but to understand those of the user, and hence usability, we need to turn to user modelling.

One approach is to consider what a user needs to know about a system in order to develop any of the U-preferences described above. First of all, they need to know what the possible modalities are. The visible presence of a keyboard may suggest to a user that typing is allowed; but will they also know that they can speak to the system? If they remain unaware of the availability of a modality, then any U-preference that includes it will not develop. Secondly, once they know that a modality is available, they will need to know what constraints the system imposes on its use. In MATIS, for example, users can specify all or part of their request by speaking. But they cannot use speech to open, close or move a window containing that request. Finally, if there is a choice between the use of different modalities, or combinations of modalities, users need to know how they should make that choice.

A technique such as Programmable User Modelling (Blandford & Young, 1993) can be used to assess the likelihood that a user will be able to acquire the particular knowledge needed to develop the appropriate U-preference. The formal aspect of constructing a PUM involves the description of the system in terms of this required knowledge, followed by the production of a runnable cognitive model that simulates the problem solving of the user. The behaviour of this model indicates whether or not a user is likely to obtain and use the knowledge appropriately. At an earlier stage of design, even the less formal process of identifying the required knowledge can help the designer understand the usability issues of alternative design solutions.

Another approach is to consider the cognitive resources a user will require in order to use or choose between the modalities included in the various U-preference sets. A general cognitive architecture such as ICS (Barnard & May, 1993) models the flow of information through different mental representations from sensation and perception, through comprehension, to action. It identifies cognitive aspects such as the influences of experience, memory requirements, and the potential for learning. The architecture also constrains the way that different sensory representations (i.e., the user's 'input' modalities) and effector representations (i.e., their 'output' modalities) can be combined. For example, for much of the time the user might have U-equivalence over spoken language, form-filling, or pointing in order to specify 'Flight From …'. But at other times, circumstances might drive her to U-assignment to a specific modality (e.g., because of difficulty in locating the information to point at, or by not being confident how to spell ‘Pittsburgh’, or how to pronounce ‘Loughborough’, and so on). Experience with a modality in some situations may lead to U-assignment in other situations where U-complementarity or U-equivalence would be more appropriate, thus undermining the flexibility of the system’s CARE properties.

5. CONCLUSION

The current definitions of the CARE properties provide a formal framework for reasoning about the design of multimodal systems. We have previously shown how the CARE properties can be used as constraints for the implementer (Nigay, 1994). In this paper, we have introduced the relationships between the system’s properties and the user’s CARE-like properties. In empirical studies, these properties could be used to structure the experimental process. Protocols can be devised according to each property and observed at different levels of granularity (e.g., goal and modality refinements). Questions raised by user modelling can serve as a useful list of concerns during design.

The formal definitions of the CARE properties cover both input and output modalities. In practice, we have applied the CARE properties to the design and implementation of MATIS which supports multimodality for input only. Our immediate short term objective is to apply the CARE properties to multimodal output.

ACKNOWLEDGEMENT

This work has been supported by project ESPRIT BRA 7040 Amodeus II. It was influenced by stimulating discussions with Phil Barnard (MRC-APU), James L. Crowley (LIFIA-IMAG), David Duke (York University) and our Amodeus partners.

REFERENCES

Barnard, P. J., & May, J. (1993). Cognitive Modelling for User Requirements. In P. F. Byerley, P. J. Barnard, & J. May (Eds.), *Computers, Communication and Usability:* Elsevier, Amsterdam.

Blandford, A. and Young, R. M. (1993) ‘Developing Runnable User Models: separating the problem solving techniques from the domain knowledge’ in J. Alty, D. Diaper and S. Guest (eds.) *People and Computers VIII*, Cambridge University Press, Cambridge.

Nigay, L. & Coutaz, J. (1993) A design space for multimodal interfaces: concurrent processing and data fusion. *InterCHI’93* ACM:New York, 172-178.

Nigay, L. (1994) Conception et modélisation logicielles des systèmes interactifs: application aux interfaces multimodales, PhD dissertation, Grenoble University.