A Syndetic Approach to Referring Phenomena in Multimodal Interaction

Giorgio P.Faconti
CNUCE Institute,
National Research Council of Italy,
56126 Pisa, Italy
G.Faconti@cnuce.cnr.it

Mieke Massink
CNUCE Institute,
National Research Council of Italy,
56126 Pisa, Italy
M.Massink@guest.cnuce.cnr.it

1 Introduction

User interfaces of many application systems have begun to include multiple devices which can be used together to input single expressions. Such interfaces (and even the whole application systems) are widely labelled multi-modal, since they use different types of communication channels to acquire information.

These emerging devices and recognition systems potentially allow users to express their intentions more naturally, in ways similar to those used by humans to communicate with each other. However, very few works have concentrated on the integration and synergistic use of multimodal input capabilities within the same system. Most systems simply take almost no account of how the different modes interact so that the interdependence of modalities contributing to information processing is not capitalized upon. Moreover, the close interaction and interdependency between input and output is still a largely unexplored area. For example, the capability of referring directly to the content of a rich multimodal presentation while formulating multimodal input requires the processing of a body of knowledge that largely extend the information content that can be conveyed by a simple pick operation.

Underlying practical use of these new technologies is the question of their suitability: are they appropriate for the tasks users need to perform, and what is their comparative ease of use?

In order to build artifacts that are to be both useful and usable, the development of interactive systems must address user-oriented requirements and accommodate different perspectives in the (formal) design process. Novel interaction techniques may interfere with the functional and task-oriented requirements that a system is intended to support. Potential conflicts between these types of requirements can be identified early in the design process through the use of appropriate specification techniques using mathematical structures able to represent perceivable elements of the system and allowing for multidisciplinary insight into the design problem.

This work describes an approach to evaluating the usability of devices that accounts for the cognitive resources needed to use a device to perform particular tasks. The framework draws its expressive power from a technique called syndetic modelling that allows the description of both the device and cognitive resources to be captured in a common representation. In this paper syndesis provides a foundation for examining the interplay occurring betwen an operator and a computer system when performing tasks involving deitic references made through speech and gestures. It is the relationship between users and systems, and the transformations that are necessary to move from one to another, that provides novel insight into usability.

2 Syndetic Modelling

The word syndesis comes from the ancient greek ($\sigma\upsilon\nu$ = together and $\delta\epsilon\omega$ = to tie), meaning to bring, to connect, to compose together. It conveys the idea of being able to reason about complex systems as a whole while keeping the capability of isolating and reasoning about their basic components at the same time.

In our case, the syndetic model of an interactive system extends the formal model of its interface with the model of the cognitive resources needed to interact with the devices. Earlier work in this direction has been using state based notations and was aiming at the exploration of this field at a high level of abstraction (Baeker and Buxton, 1987; Chan et al., 1984). In other approaches theoretical models originating from psychology have been used in an *indirect* way, see for example (Card et al., 1990; Fitts, 1954). We deviates from those early approaches by using cognitive models in a *direct* way within the design and specification process and find our justification for such an approach in that the factors that affect

usability depend on psychological and social properties of cognition and work, rather than on abstract mathematical models of programming semantics.

Although in principle any cognitive theory might be adopted, we address one particular cognitive model, Phil Barnard's Interacting Cognitive Subsystems or shortly ICS (Barnard and May, 1993; Barnard and May, 1994). We formally model aspects of this theory in such a way that it can be combined with a traditional system specification. The formal model of the system provides few insights into the usability of its interface as well as the formal model of the user derived from some psychological theory supports general claims about the user's cognitive processes but not about the effective use of cognitive resources in a given context. By combining both of them in a syndetic model we can reason about how cognitive resources are mapped onto the functionality of the system.

Within this approach, we consider an abstract view of the flow of information between devices, users and system. To facilitate precise description and modelling at this level, we make use of a specification notation in which the various components (device, system and user) are modelled as interactors. The concept of an interactor has been described in detail elsewhere, for example (Duke and Harrison, 1993; Faconti and Paterno, 1990). Briefly, an interactor is an object-like entity with an internal state, a presentation through which parts of the state (called percepts) can be perceived by a user, and actions - either user or system initiated - that bring about changes to the state. Interactors have been described using a number of formal notations including Z, LOTOS and MAL (Modal Action Logic), and it is the last of these that is used here. Briefly, MAL (Ryan et al., 1991) is a typed first-order logic that extends the predicate logic with an additional operator. For any action 'A' and predicate 'P', the predicate '[A] P' means that after the action A is performed, P must hold.

Interactors can describe the logical and physical components of an interactive system, but by themselves give little direct insight into how a user might or might not be able to use the system. This is a problem, as many of the developments in interactive systems that can benefit from use of abstract models also depend critically on human abilities to process information. Syndetic models (Duke, 1995; Duke et al., 1995; Faconti and Duke, 1996) address this problem by expressing the behaviour of computing and cognitive systems within a common framework that supports reasoning about the conjoint system. Clearly, the 'computer' component of a syndetic model is determined by the system being represented, but for the cognitive side there is a range of models to choose from, each emphasising different aspects of human information processing. The approach that we have adopted for syndetic modelling is called Interacting Cognitive Subsystems, or ICS, and is summarised in Section 3. Importantly, ICS operates in terms of resources and information flow at a level of abstraction that is commensurate with that used to describe interactors.

3 Interactive Cognitive Subsystems (ICS)

ICS is a comprehensive model of human information processing that describes cognition in terms a collection of sub-systems that operate on specific mental codes or representations. Barnard and May identify two major aspects of ICS: a theory of representation and a theory of information flow. Interestingly, the two kind of theories can be related respectively to abstract data types and state based specifications, and to process algebraic and data flow approaches in computer science. The work on syndetic modelling has concentrated only on the capturing the theory of information flow and on exploring problems by reasoning about information flow. The area of the representation of the mental codes has not yet been explored from a formal perspective. Recently, the ICS project at MRC Applied Psychology Unit in Cambridge and at the Departments of Psychology of University of Sheffield and Copenhagen has developed a systematic treatment of visual structures (May et al., 1995; May et al., 1997) that will be part of our future research.

3.1 Information flow in ICS

ICS represents the human information processing as a highly parallel organization with a modular structure composed of nine sub-systems. Although specialised to deal with specific codes, all subsystems have a common architecture, shown in Figure 1.

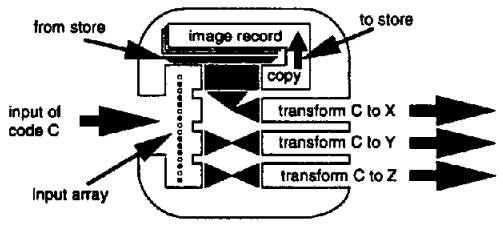

Figure 1: Generic structure of an ICS sub-system.

These subsystems can perform two kinds of operation upon the representations that they receive at

an input array. They can copy the representation directly into the image record, which acts as a memory local to each subsystem, and they can transform the information into another mental representation and pass it through a data network to other subsystems. The transformation processes within each subsystem are independent and can work in parallel.

The representations that can be output by a subsystem are limited by the informational content of the representations that it operates upon; that is, a subsystem cannot produce output in every representation. Moreover, any one transformation process can only operate upon a single coherent data stream at one time. That is, it can only operate upon one representation, and can only produce one output representation.

If the incoming data is incomplete, a subsystem can augment it by accessing the image record. Coherent data streams may be blended at the input array of a subsystem, with the result that a process can transform data derived from multiple input sources in one step. This balances the output limitation.

The nine subsystems are further distinguished depending on their functionality as:
Sensory subsystems

VIS visual: hue, contour etc.
AC acoustic: pitch, rhythm etc.
BS body-state: proprioceptive feedback

Meaning subsystems

IMPLIC implicational: holistic meaning
PROP propositional: semantic relations

Structural subsystems

OBJ object: mental imagery, etc.
MPLmorphonolexical: lexical forms, etc.

Effector subsystems

ART articulatory: subvocal rehearsal, etc.
LIM limb: motion of limbs, eyes, etc

The nine subsystems acts effectively as communicating processes running in parallel as shown in Figure 2.

The overall behaviour of the cognitive system is governed by a number of principles, most of which are out of the scope of this paper. Here, we will address only those configurations that are relevant to interact with the system described in the previous section. Configurations are the way in which ICS resources are deployed at a point in time to perform a cognitive task. Complex configurations can be constructed from elementary, partial ones, and if an information flow can be constructed, then it is a legal configuration, subject to three constraints. The first one is that no process can appear more than once in a configuration. The second constraint is that the order of cyclical flows within the configuration is not important. Finally, although any one of the sensors or effectors may be missing, if all sensors or effectors or both are missing in a configuration there must be a central flow. In other terms, input alone is meaningless and no output can be generated without either input or central activity.

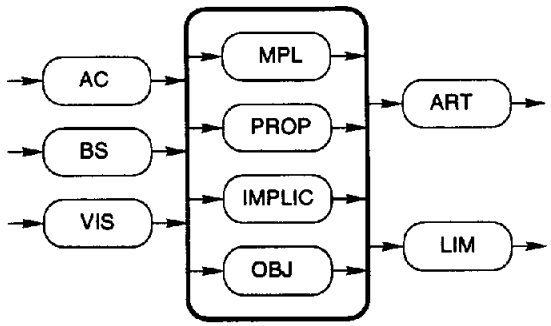

Figure 2: Architecture of ICS.

3.1.1 A formal account of ICS

The key observation underlying syndetic modelling is that the structures and principles embodied within ICS can be formulated as an axiomatic model in the same way as any other information processing system. This means that the cognitive resources of a user can be expressed in the same framework as the behaviour of computer-based interface, allowing the models to be integrated directly. To begin this process, we define some sets to represent those concepts of ICS that will be used here. Here and elsewhere in this document we will make use of the Z notation (Spivey, 1982) to define data types; much of this is based on common mathematical conventions for sets and relations, for example '$\times$' for cartesian product and '$\mathbb{P}$' for power set.

[sys] : ICS subsystems
[repr] : representations
$\mathrm{tr} == \mathrm{sys} \times \mathrm{sys}$

Representations consist of basic units of information organised into superordinate structures. Coherence of units depends on several issues, including the timing of data streams, that will not be addressed here. Instead, coherence is captured abstractly in the form of an equivalence relation over representations:

$\_ \approx \_ : \mathrm{repr} \leftrightarrow \mathrm{repr}$

In describing ICS it is also useful to discuss the representations that are being delivered as part of a particular data stream. We therefore introduce a

further set, code, whose elements are representations that have been labelled by the subsystem in which they were generated. Representations from or to the outside world are tagged with '*':

$code == repr \times sys$

In general we will write R_{sys} for the code (R, sys), and ':src-dst:' for the transformation (src, dst).

The state of the ICS interactor captures the data streams involved in processing activities and the properties of the streams such as stability and coherence which define the quality of processing, or in other words, user competence at particular tasks. The sources of data for each transformation is represented by a function 'sources' that takes each transformation 't' to the set of transformations from which 't' is taking input. In general only a subset of transformations are producing stable output, and this set is defined by the attribute 'stable'. The codes that are available for processing at a subsystem are identified by a relation _@_, where 'c@s' means that code 'c' is available at subsystem 's'.

interactor ICS
attributes

sources	:	$tr \rightarrow \mathbb{P}\, tr$
stable	:	$\mathbb{P}\, tr$
_@_	:	$code \leftrightarrow sys$

As not all representations are coherent, only certain subsets of the data streams arriving at a system can be employed by a process to generate stable output. The set 'coherent' contains those groups of transformations whose output in the current state can be blended. If the inputs to a process are coherent but unstable, the process can still generate a stable output by buffering the input flow via the image record and thereby operating on an extended representation. However, only one process in the configuration can be buffered at any time[1], and this process is identified by the attribute 'buffered'. The configuration itself is defined to be those processes whose output is stable and which are contributing to the current processing activity.

coherent	:	$\mathbb{P}\,\mathbb{P}\, tr$
buffered	:	tr
config	:	$\mathbb{P}\, tr$

Four actions are addressed in this model. The first two, 'engage' and 'disengage', allow a process to modify the set of streams from which they are taking information, by adding or removing a stream. A process can enter buffered mode via the 'buffer' action. Lastly, the actual processing of information is represented by 'trans', which allows representations at one subsystem to be transferred by processing activity to another subsystem.

actions

engage	:	$tr \times tr$
disengage	:	$tr \times tr$
buffer		
trans		

The principles of information processing embodied by ICS are expressed as axioms over the model defined above. Axiom 1 concerns coherence, and states that a group of processes are coherent if and only if they have the same kind of output (in the code of the system 'dest') and that the representations produced by the processes and therefore available at 'dest' are themselves coherent.

axioms

1 $\forall trs : \mathbb{P}\, tr \bullet trs \in coherent$
$\Leftrightarrow$
$\exists dest : sys \bullet$
$$\left(\begin{array}{l} \forall s, t : sys \bullet :s\text{-}t: \in trs \Rightarrow t = dest \\ \wedge \\ \forall s, t : sys;\ p, q : repr \bullet \\ \left(\begin{array}{l} :s\text{-}dest: \in trs \wedge p_s @ dest \\ \wedge \\ :t\text{-}dest: \in trs \wedge q_t @ dest \end{array} \right) \Rightarrow p \approx q \end{array} \right)$$

The second axiom is that a transformation is stable if and only if its sources are coherent, and either it is buffered or the sources are themselves stable. A configuration then consists of those processes that are generating stable output that is used elsewhere in the overall processing cycle.

2 $t \in stable \Leftrightarrow sources(t) \in coherent \wedge (t = buffered \vee sources(t) \subseteq stable)$

3 $t \in config \Leftrightarrow (t \in stable \wedge \exists s \bullet t \in sources(s))$

A process will not engage an unstable stream (axiom 4). If its own output is unstable, it will either engage a stable stream, disengage an unstable stream, or try to enter buffered mode (axiom 5). The remaining axioms (5-7) define the effects of the three actions.

4 $\mathbf{per}(engage(t, src)) \Rightarrow src \in stable$

5 $t \notin stable \Rightarrow$
$$\left(\begin{array}{l} \exists s \bullet s \in stable \wedge s \notin sources(t) \wedge \\ \mathbf{obl}(engage(t, s)) \\ \vee \\ \exists s \bullet s \notin stable \wedge s \in sources(t) \wedge \\ \mathbf{obl}(disengage(t, s)) \\ \vee \\ \mathbf{obl}(buffer(t)) \end{array} \right)$$

The effects of the buffer, engage, and disengage actions are straightforward and are given by axioms 6-

[1] This is actually a simplification for the purposes of the paper.

8.

6 [buffer(t)] buffered = t

7 sources(t) = S ⇒
[engage(t, s)] sources(t) = S ∪ {s}

8 sources(t) = S ⇒
[disengage(t, s)] sources(t) = S − {s}

The remaining two axioms define the effect of information transfer. Axiom 9 is the 'forward' rule: if a representation is available at a subsystem, then after trans a suitable representation will be available at any other subsystem for which the corresponding process is stable. Conversely, if after trans some information were to become available at a subsystem (dest), then there must exist some source system such that the information is available at the source, and the corresponding transformation is stable.

9 p_x@src ∧ :src-dst: ∈ stable ⇒ [trans] p_{src}@dst

10 (∃ p : repr; src, dst : sys • [trans] p_{src}@dst) ⇒
∃ x : sys • p_x@src ∧ :src-dst: ∈ stable

3.2 The structure of mental representations

Most of the formal account of ICS given in the previous section relies on an understanding of representations and of their structure.

In (May et al., 1997) the process of perception is described as one of structuring the sensory information that we receive from objects in the environment so that we can interact with them. The details about the structure of objects and their inter-relations are not explicitly contained in the sensory information. It must be interpreted by combining this information with knowledge about the world, which we have learnt through our experience of interacting with it.

Computer displays are like the rest of the world in this respect. Consequently designing a computer display is all about choosing the form of objects and arranging them so that they are perceived and dealt with by the user of the computer. Different arrangements of the same set of forms may lead to different structuring of objects' representations. This may result in different performances of a particular task by the user.

When we look at a visual scene, the features, colors and textures in the sensory information group together to form objects. If we look closely at an object, we can see that it has also a structure and may be composed by other objects. We can see the world at different scales, from a global level, down to many levels of details. For example, figure 3 can be seen as a computer display with objects in it. Focusing the attention toward a particular object we may see either a window or a cursor and so on. This hierarchy can be represented as a structure diagram, as in figure 4, where the horizontal groupings are sets of objects at different levels of the visual structure.

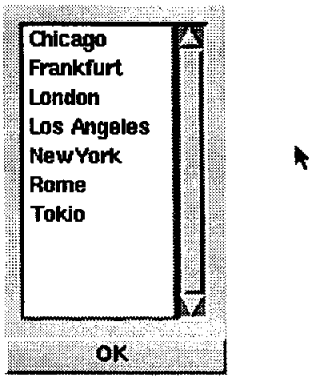

Figure 3: Objects within a computer display.

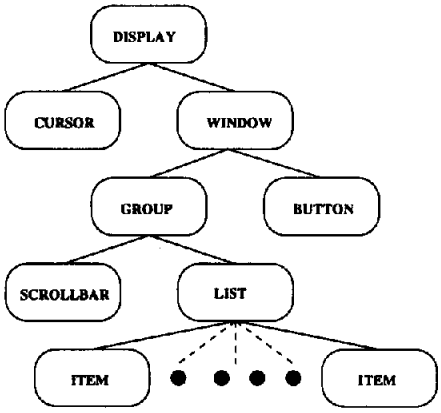

Figure 4: Information Structure.

What we perceive at a given moment is limited by the level at which we analyse the scene. For example, a test made with figure 3 on a number of our colleagues revealed that the totality of them sees a 'list of cities that can be scrolled'. Clearly, this information is the result of an interpretation of the raw sensory data obtained from the eyes and enriched by a set of mental processes that convert the visual representation into an object one to which a semantic information is further added.

What it is important to notice is that the attention has been focused on the 'list' node in the structure, that to reach that node one might have searched through it, and that 'list' is related to 'scrollbar'. According to (May et al., 1997) we say that 'list' is the *psychological subject* being attended, 'scrollbar' (i.e. objects in the same group of the psychological subject) forms its *predicate*, and 'cities' (i.e. the sub-structure rooted at the psychological subject) form its *attribute*. The attention can be easily

moved towards one of the predicates of the subject by swapping the subject-predicate relation. Diverting the attention to a far object in the structure requires much more cognitive load since it implies the traversal of larger parts of the structure.

Clearly we are describing a 'static' situation where the persons were explicitly asked to perform only a recognition task. In dynamic (real case) situations the same sensory information is interpreted to perform different tasks either in a sequence or in parallel. For example, to move the cursor over an item (i.e. a city name) one must establish a relation between the cursor and the item that requires a reworking of the structure. This can be described as defining a ghost object to which both the cursor and the item are rooted. The ghost is maintained until the cursor-item relation is needed to perform the required task and hides the previous structure for that period of time as shown in figure 5. During this period the objects in hiding cannot part of the psychological subject. Designing presentations leading to stable structures over tasks greatly increases the ease of the interaction by reducing the cognitive load necessary for the restructuring of structures.

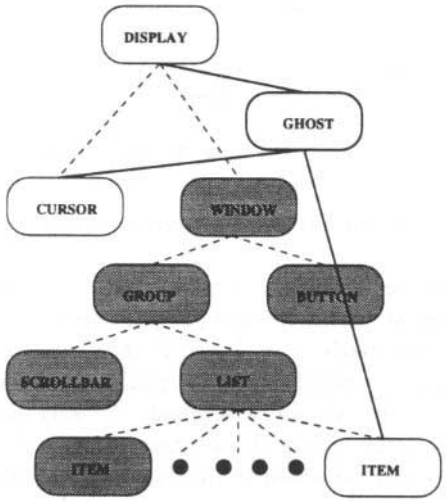

Figure 5: Ghost node within the information structure.

This reasoning leads to add a further axiom to the ICS theory. Two transformation processes within the same subsystem can act in parallel over the same representation or over two representations such that one is not a sub-structure of the other (they are *disjoint*). Disjonction is captured abstractly in the form of a relation over code:

$\_ \Phi \_ : \text{code} \leftrightarrow \text{code}$

11 $(\exists\, p, q : \text{repr};\ src, d1, d2 : sys \bullet$
$[\text{trans}]\ p_{src}@d1 \wedge q_{src}@d2 \wedge d1 \neq d2) \Rightarrow$
$p \approx q \wedge p_x@src = q_y@src \vee$
$p_x@src\ \Phi\ q_y@src$

3.2.1 Levels of mental representations

In the previous section we have seen that sensory information is interpreted in order to build structured mental representations. The interpretation requires the participation of several subsystems that are deployed in a configuration. The understanding of the structure in figure 4 is given by that the sensory information from eyes forms a visual representation made of colours and the likes that gives rise to the configuration represented in figure 6.

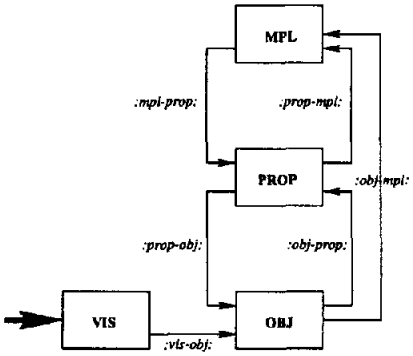

Figure 6: Reading configuration.

A mental process (VIS) transforms (:vis-obj:) it into an object representation that involves the structuring of sensory data into objects, and the grouping together of those objects. This new representation can be interpreted by another mental process (OBJ) and transformed (:obj-prop:) to produce a more abstract representation at propositional level in which objects are identified and related. At this point a third transformation (:prop-obj:) takes place at the propositional subsystem (PROP) that feeds back information about object structure. After this tranformation the object structure that is perceived is a blend of information from propositional and visual sources. For this to take place, a number of conditions must be met according to the formal ICS theory, such as:

$\{\text{:prop-obj:}, \text{:vis-obj:}\} \in \text{coherent}$
$p_{prop}@obj \wedge q_{vis}@obj \Rightarrow q \approx p$
$\{\text{:prop-obj:}, \text{:vis-obj:}, \text{:obj-prop:}\} \subseteq \text{stable}$

The configuration deployed so far doesn't justify that the items in the list are recognized as cities. In order to do this the objects' structure must be made available to the morphonolexical system (MPL) as a structured representation of sound. Consequently,

the :obj-mpl: transformation operates in parallel with the :obj-prop: one on the same code and produces a morphonolexical representation that is equivalent to the one sent to the propositional susbsystem. The morphonolexical susbsystems transforms (:mpl-prop:) the representation into propositional code that is blended to the one produced directly by the object susbsystem. At the propositional system the :prop-mpl: transformation is activated in parallel with the :prop-obj: that feeds back semantic information to the morphonolexical system and enrich the object structure by blending with the object source. Again this requires that some additional properties are satisfied in the ICS theory, such as:

$\{\{$:obj-prop:, :mpl-prop:$\}$,
$\{$: prop $-$ mpl :, : obj $-$ mpl :$\}\} \in$ coherent
$p_{obj}@prop \wedge q_{mpl}@prop \Rightarrow q \approx p$
$p_{obj}@mpl \wedge q_{prop}@mpl \Rightarrow q \approx p$
$\{$:obj-mpl:, :mpl-prop:, :prop-mpl:$\} \subseteq$ stable

4 The cognitive configuration for deitic reference

The configuration described in the previous section can be defined as the *reading* one. In fact, it might be noticed that once the object representation is transformed by :obj-mpl: and made available to the morphonolexical subsystem it is also ready to be spoken by the articulatory system after an :mpl-art: transformation. This *read aloud* configuration is obtained by adding the :mpl-art: and the :art-speech: transformations to the reading configuration so that

$\{$:mpl-art:, :art-speech:$\} \subseteq$ stable

A similar reasoning can be applied to the object subsystem in the sense that once the object structure is formed, the :obj-lim: tranformation can generate the limb code equivalent to the object representation so that (for example) the hand operates the currently selected psychological subject. The new configuration is obtained by imposing that

$\{$:obj-lim:, :lim-hand:$\} \subseteq$ stable

Together with the described configuration two feedback loops exist involving the body-state subsystem which is a source of sensory information. This information represents sensations that our body detects from tasting, touching and smelling as well as information from internal sensations such as the position of our arms and legs and the state of our muscles.

In our case the body state transforms two disjoint representations from an interpretation of the hand position and muscle state, and of the state of the vocal muscles. The information at this level of representation is important to co-ordinate our physical actions because they enrich the limb and articulatory representations by blending with those produced by the object and the morphonolexical subsystems. Clearly, the followings must hold:

$\{$:bs-lim::bs-art:$\} \subseteq$ stable
$p_{bs}@art \wedge q_{bs}@lim \Rightarrow p_*@bs \; \Phi \; q_*@bs$

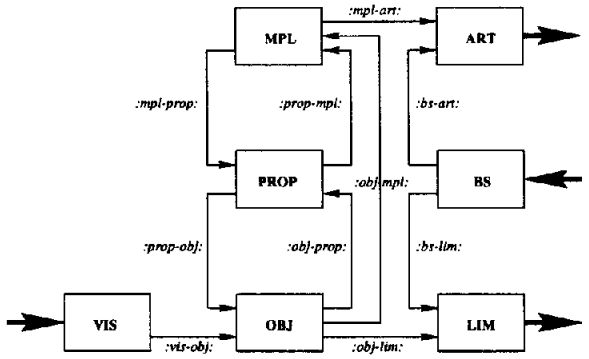

Figure 7: Speech and gestures configuration.

The final configuration describing the cognitive view of performing a deitic reference by speech and gestures is shown in figure 7. In the following we will refer to the configuration as deixis – Conf.

5 Description of the system interface

From the system perspective, the problem can now be formulated as the spefication of a presentation that allows the speech and gesture configuration of ICS to be naturally deployed when making use of deixis.

In principles, the devices we could use to implement an interface supporting deixis range from traditional tablets to data gloves, from cameras to video recorders and players, from speakers to microphones, from flat to head mounted displays with stereoscopic views, and many others. Here we will compare two systems respectively built from a display and a mouse, and a display equipped with a touch screen. The comparison can be easily extended to the case of devices with similar caracteristics with respect to the addressed task such as a tablet instead of the mouse, and a data-glove for the touch screen.

5.1 Display and mouse based interface

The most common and widespread graphical device is the 2D mouse, a physical device equipped with two transducers able to measure the distance between a current position and a next point along two axes and with a number of buttons (usually from one to

three). The buttons have little value for the purposes of this paper, and are disregarded. The mouse can be described by a very simple interactor, where the type 'RelPos' represents *relative* positions, i.e. offsets.

interactor Mouse
attributes
mouse : RelPos
actions
[bs] operate : RelPos
axioms
1 [operate(δ)] mouse = δ
2 [[operate]] in [[Mouse]]

The Mouse interactor describes the state space of the device as a coordinate defining the distance of the current position from the previous one along two coordinate axis (RelPos == delta−xMouse × delta−yMouse). The [bs] decoration of the 'operate' action means that the device is sensed by the body-state subsystem when it is used, and the notation [[...]] is used to refer to the perceivable aspect of an attribute, interactor or action.

While the mouse can be used as a pure input device, it is usually coupled with a cursor that provides the feedback of the current position in the display space (DispPos). The cursor is an object amongst the others of type Obj in a display whose position is related to the mouse by a coordinate transformation. Consequently, we explicitly distinguish the cursor in specifying a display interactor.

interactor MDisplay
Mouse
attributes
[vis] objects : $\mathbb{P}$Obj
[vis] cursor : Obj
location : Obj → DispPos
transform : RelPos → DispPos
_ relate _ : DispPos ↔ DispPos
actions
render
axioms
1 cursor ∈ objects
2 location(cursor) = P ∧ mouse = δ ⇒
[render]location(cursor) = P + transform(δ) ∧
mouse = (0, 0)
3 [[objects]] in [[Display]]
4 o ∈ objects ⇒
(cursor relate o ⇔
location(cursor) = location(o) ∧
[[cursor, o]] in [[Display]])

Objects are located in the display. The cursor location is computed by transforming the current mouse movement at the next refresh of the screen (action 'render'). An object in the display is related to the cursor when it has the same position. The [vis] decoration indicates that the objects in the display are visually perceivable.

5.2 Touch-screen based interface

If we plan to use a touch-screen display to build our interface, there exist only one device, namely the display. In contrast with the mouse-display pair, the [bs] and [vis] percepts apply to the same attributes.

interactor TDisplay
attributes
[vis]
[bs] objects : $\mathbb{P}$Obj
location : Obj → DispPos
actions
[bs] operate : DispPos
axioms
1 [[objects]] in [[Display]]

6 Building the Syndetic Model

The syndetic model of device interaction is created by introducing both the user and system models into a new interactor and then defining the axioms that govern the conjoint behaviour of the two agents. A new attribute (*goals*) is used to 'contextualise' the generic ICS model to the task of making a deitic reference as set of pairs Obj × Operation. Here, a more realistic approach might be to describe a class of desired or acceptable displays. However, it would add little to the analysis.

interactor MDeixis
MDisplay (alternatively TDisplay)
ICS
attributes
goals : $\mathbb{P}$(Obj × Operation)

The configuration must be set to deixis−Conf and the (*goals*) attribute is initialized. For the goal to be achieved we locate the buffer to the propositional subsystem to revive the :prop-obj: transformation.

axioms
1 deixis-Conf ⊆ config ∧ buffered = :prop-obj:
2 []goals = (item, read);
((item, speak) || (item, locate))

In initializing the goals we use the action prefix ';' notation to indicate sequentiality and '||' to indicate parallel composition.

7 Analysis

We will examine the above specified model informally, since there is not space to conduct a full for-

mal analysis. The interested reader may address the referenced papers on syndesis for a more deep understanding. Here we will show directly the result of the analysis and will make comments on it.

To satisfy the first sub-goal (item, read), the object subsystem receives coherent representations from :prop-obj: and :vis-obj: that are in its sources. They must be also stable and coherent so that their representations are blended. The enriched representation is tranformed by the object system into propositional, morphonolexical and limb representations. Since the goal is to read, the psychological subject becomes an entry in the list. The morphonolexical system can operate on this representation in order to find its related sound structure. Similarly the propositional system revives it through its buffer to both morphonolexical and object systems enriching their representations.

In the case of the MDisplay system, which uses the mouse, the information transmitted by the object system is of little use for the limb system. In fact, the cursor is far from the psychological object in the representation structure. Consequently, the information from the body-state which 'feels' the mouse troough the 'ooperate' action and the one from the object system cannot be blended leading to buffering. However the buffer is already allocated and consequently the stream is disengaged leading to a change of the configuration.
In the case of the TDisplay model, which makes use of the touch screen, the same stream resulting from the :obj-lim: transformation is relevant to the limb system since it blends with the information arriving from the body-state. In is interesting to note that in this second case the movement of pointing to an item starts before the same information is processed by the articulatory system for speaking. This is confirmed by experiments in the field of cognitive psychology.

After one cycle of processing of the goal by all the involved subsystems, the propositional system removes the first part of the goal and starts satisfying the two parallel tasks of speaking and gesturing by sending representations again to the morphonolexical and object subsystems. At the morphonolexical level this representation blends naturally since all the information was already available for specking and it can be passed directly to the articulatory subsystem. At the object level the new representation blend with the information stream from the visual subsystem.

In the case of the MDisplay system, the ghost node of figure 5 is built and sent to the propositional system for semantic checking. Only after a further loop between the propositional and the object systems this information is sent to the limb system where it can now be blended with the body-state information to perform the pointing gesture. However, at this time the articulatory system has already directed the speech of the referred word. Consequently, in this case the speech and locate actions cannot occur in parallel but are performed in a sequence.
In the case of the TDisplay model, the limb system has already started to locate the item within the screen so that the operation can continue in parallel with the articulatory system and synchronize through the body state.

The result is extremely interesting when related to previous works carried on the process of fusion of information within multimedia systems.

At University of Grenoble, CLIPS, they have developped an original algorithm, known as the 'melting pot', to support deixis within the Matis system. Matis is a Multimodal Airline Travel Information System supporting several combinations of modalities to formulate queries against a flights data base. The melting pot algorithm is built around the intrinsic uncertainty found in relating mouse events and spoken words, the authors have directly experimented in building the system. The practical consequence is that the algorithm is noon-deterministic.

Our woork clearly gives a motivation for this. In (Faconti et al., 1996) the fusion process is described at a high level of abstracion. It defines a system architecture of fusion and a class of algorithms which the melting pot is one instance of. The work is in line with the findings of this paper suggesting that a non-deterministic fusion algorithm can be developped based on exact temporal windows within which pointing events may occur. These temporal windows are defined by the limb and articulary subsystems processes within ICS and can be captured by the system speech recognizer.

8 Conclusions

Traditional approaches to evaluating or comparing input devices have focussed either on the logical behaviour of the device, or ergonomic aspects of its use. This paper has presented a framework that allows analysis of the *cognitive* ergonomics of interaction, in terms of the mental resources needed to utilise a particular device for a specific task. We have used the model to present a systematic account of the diffences between mouse and touch screen. The example was chosen for familarity, rather than for novelty. Hoowever, the approach is one that can be extended to rather more sophisticated and problematic techniques.

One argument raised against the wider use of syndetic modelling for human factors evaluations is the level of formality involved. This is a reasonable concern, and there are two responses. The first is that the work on syndesis carried out so far has been primarily concerned with establishing its feasibility as a model for explaining interaction, rather than as a practical tool for industrial use. We are now beginning to explore means by which the level of formality can be tamed, both by supporting development of formal models with software tools, or by encapsulating the technique within a tool to support scenario-driven analysis of interaction.

The second response to concern about formalism is that the complexity of modern interfaces, and the subtle demands that they place on users' cognitive abilities, calls for an expressive and analytically powerful method for modelling and evaluation. Such a method needs to be able to span both the user and the system, in order to capture the interplay between the information available from the system, the actions that can be taken, and the tasks and knowledge of the user. We are not advocating syndetic models as a replacement for other design representations. There is an inherent trade-off between the power and generality of a notation (Blanford and Duke, 1996), and there are important issues, for example based on social factors or domain requirements, for which syndetic models are either inappropriate or inadequate. Likewise however, syndesis brings considerable analytical power and authority (in the form of the underlying cognitive theory) to bear on problems whose complexity makes the use of less formal design techniques problematic.

References

R.M. Baecker and W. Buxton, editors. 1987. *Readings in human-computer interaction: A multidisciplinary approach.* Morgan-Kaufmann.

P.J. Barnard and J. May. 1993. Cognitive modelling for user requirements. In P.F. Byerley, P.J. Barnard, and J. May, editors, *Computers, Communication and Usability: Design Issues, Research and Methods for Integrated Services*, North Holland Series in Telecommunication. Elsevier.

P.J. Barnard and J. May. 1994. Interactions with advanced graphical interfaces and the deployment of latent human knowledge. In *Eurographics Workshop on Design, Specification and Verification of Interactive Systems*, pages 15–49. Springer.

A. Blandford and D.J. Duke. 1996. Integrating user and computer system concerns in the design of interactive systems. *IEEE Transactions on Software Engineering.*

S.K. Card, J.D. Mackinlay, and G.G. Robertson. 1990. The design space of input devices. In *Proc. of CHI'90.* ACM Press.

S.K. Card, J.D. Mackinlay, and G.G. Robertson. 1990. A semantics analysis of the design space of input devices. *Human-Computer Interaction.*

D.J. Duke. 1995. Reasoning about gestural interaction. *Computer Graphics Forum*, 14(3):55–66. Conference Issue: Proc. Eurographics'95, Maastricht, The Netherlands.

D.J. Duke, P.J. Barnard, D.A. Duce, and J. May. 1995. Systematic development of the human interface. In *APSEC'95: Second Asia-Pacific Software Engineering Conference*, pages 313–321. IEEE Computer Society Press.

D.J. Duke and M.D. Harrison. 1993. Abstract interaction objects. *Computer Graphics Forum*, 12(3):25–36. Conference Issue: Proc. Eurographics'93.

G.P.Faconti, M. Bordegoni, K.Kansy, T. Rist, P. Trahanias, and M.D. Wilson. 1996. Formal Framework and Necessary Properties of the Fusion of Input Modes in User Interfaces *Interacting with Computers*, Vol. 8(2), pp. 134-161, Elsevier Science B.V.

G. Faconti and D. Duke. 1996. Device Models. In F. Bodart, and J. Vanderdonckt , editors, *Design, Specification and Verification of Interactive Systems*, pages 73–91. Springer-Verlag.

G. Faconti and F. Paterno'. 1990. An approach to the formal specification of the components of an interaction. In C. Vandoni and D. Duce, editors, *Eurographics 90*, pages 481–494. North-Holland.

P.M. Fitts. 1954. The information capacity of the human motor system in controlling amplitude of movement. *Journal of Experimental Psychology*, 47:381–391.

P. Chan, J.D. Foley, V.L. Wallace. 1984. The human factors of computer graphics interaction techniques. *Computer Graphics and Applications*, 4(11).

J. May, S. Scott, and P. Barnard. 1995. *Structuring Interfaces - a psychological guide.* Eurographics'95 Tutorial Notes. European association for Computer Graphics, Geneva.

J. May, S. Scott, and P. Barnard. 1997. *Modelling multimodal interaction: A theory-based techniques for design, analysis and support.* INTERACT'97 Tutorial Notes. Available also at http://www.shef.ac.uk/~pc1jm/guide.html

M. Ryan, J. Fiadeiro, and T. Maibaum. 1991. Sharing actions and attributes in modal action logic. In T. Ito and A.R. Meyer, editors, *Theoretical Aspects of Computer Software*, volume 526 of *Lecture Notes in Computer Science*, pages 569–593. Springer-Verlag.

J.M. Spivey. 1992. *The Z Notation: A Reference Manual.* Prentice Hall International, second edition.