Electronic Notes in Theoretical Computer Science 43 (2001)
URL: http://www.elsevier.nl/locate/entcs/volume43.html

Syndetic Modelling: Computer Science Meets Cognitive Psychology

D.A. Duce [1]

School of Computing and Mathematical Sciences
Oxford Brookes University
Oxford, U.K.

D.J. Duke [2]

Deptartment of Mathematical Sciences
University of Bath
Bath, U.K.

Abstract

This paper discusses an approach to combining user and system models within a common mathematical framework, to yield an integrated view of human-system interaction. Aspects of systems that affect usability can be described and reasoned about in terms of the conjoint behaviour of user and computer. The paper outlines the basic ideas behind the approach through a case study. The paper then reviews what has been achieved so far in this endeavour and concludes with a discussion of prospects and open questions.

1 Introduction

This paper describes the origins of syndetic modelling, gives an overview of what has been achieved so far and concludes with some thoughts on future directions. The work originated in an EU-funded project, Amodeus II (Esprit Basic Research Action 7040) which brought together researchers from a number of disciplines, including computer science and cognitive psychology, with a common interest in developing methods for designing and analysing interactive systems [5]. In the main the Amodeus project looked at integrating the results of HCI modelling. Such an approach, where the underlying theory is used only by experts in a limited domain, and is then discarded, may be reasonable in certain design contexts, where the technology or human aspects are relatively well understood. This is not the

[1] Email: daduce@brookes.ac.uk

[2] Email: D.Duke@bath.ac.uk

case with novel interface technologies such as haptics or vision tracking. Equally, separating modelling theory from recommendations leaves the HCI designer with no sound or systematic basis for discovering the cause of usability problems, or a framework for exploring alternatives. The work described here is significant in that it starts from the premise that the underlying theory should be preserved and be visible in order to support reasoning about the behaviour of interactive systems [7].

The behaviour interactive system can be thought of as the *conjoint* behaviour of the computer system and the system's user. Within the Amodeus project, the research groups represented tended to be champions of techniques for modelling either the computer system or the user, and there had been relatively little attempt at fitting such models into a broader framework that would encompass behaviour of both computer system and user. One exception to this is the Interaction Framework [8]; however, like other work on modelling interaction, for example [25], it describes interaction in terms of what can be observed at the interface; the internal capabilities and constraints of the human and computer that give rise to this behaviour are not represented and therefore cannot be used to argue about interaction. In syndetic modelling, mathematical structures are used as a framework for both computer system modelling and the representation of a cognitive theory. It provides a common language that records both interaction between human and computer, and the theories that describe how the individual agents operate. The name was chosen as 'syndetic', which is derived from a Greek word 'sundeo', means 'bring together'.

If theoretically-based modelling is to provide designers with insight into how to *improve* an artefact, then the theory underlying the analysis must be brought forward so that it can be inspected to reveal why some design issue is problematic and what modifications of the design would address the issue. Such an integration of user and system representations is necessary to describe (let alone resolve) certain design issues, as illustrated by Duke [13]. Duke describes a gesture-based system for manipulating 3D objects. Using a model or theory of presentations, images displayed on the screen might be described as a collection of percepts, for example a cube, cone and sphere. Requirements involving the presentation can be stated, for example that the presentation of the sphere should be 'hidden' behind that of the cube. Such a constraint makes no mention of what information the designer of an interface expects or intends the user to perceive. Without some record of intended interpretation, it is difficult to argue whether an interaction or presentation technique is appropriate. For example, the designer of a 3D object editor needs to consider whether the presentation of a system will allow the user to construct an appropriate 'mental model' of the relevant state. In order to express and reason about such issues, the designer will need some appropriate representation of human cognitive abilities.

Modern user interfaces are beginning to employ a range of modalities, including graphics, sound and haptics. Such interfaces imply that the user is potentially processing several streams of information concurrently. To describe and analyse such interfaces, it is even more important to be able to represent the human information processing system that is operating on the information streams delivered by the computer system. Syndetic modelling takes a first step towards addressing the challenges posed by such innovative combinations of interface technology.

2 An Example

The basic idea of syndetic modelling is to bring together an approach to computer system modelling and an approach to cognitive modelling within a common framework. Syndetic modelling as presented here is based on a commitment to a particular system modelling approach (interactors) and a particular approach to describing human cognition (ICS - Interacting Cognitive Subsystems). There is also a commitment to a particular mathematical framework, in this case Modal Action Logic (MAL) [20,28,22] (or to be more precise, an early version of MAL). These choices are to some extent arbitrary. They were motivated by the context in which syndetic modelling was developed (the approaches represented in the Amodeus project) and a sense that these pieces could be drawn together in a useful way. Other authors [9] have explored the use of other formal notations - LOTOS and the interval temporal logic Mexitl, for describing aspects of ICS. The choice of an appropriate mathematics is a question to which we return later in this paper.

The key components of syndetic modelling will be introduced through an example case study.

2.1 MATIS

The case study is based on an experimental interface, MATIS [26,27], that was developed to explore the use of multimodal input. The domain of MATIS is flight information; the systems allows a user to plan a multi-stage journey by completing ‘query’ forms that can be used to search a database for a matching flight. The forms can be completed using multiple modalities, including direct manipulation, keyboard entry, and speech. These can be employed either individually or in combinations. A feature of the system is support for *deictic* references, in which, for example, spoken informaiton can be combined with gesture via a mouse to produce a single command for an application. The example in Figure 1 shows a user combining spoken natural language with mouse-based gesture to fill in the second query template. On the left hand side the user has begun to speak a request that contains a deictic reference, “this city”, which is resolved when the user clicks on a field containing a city name; the right hand side of Figure 1 shows the query form after the system has interpreted the user’s input.

A detailed mathematical model of MATIS will not be developed here; this has been done elsewhere [17]. The model of MATIS developed here is minimal, being sufficient just to illustrate the approach, and the questions that arise concerning usability.

The specification uses the notion of an interactor [16,19]. An interactor consists of an internal state representing some facet of the application domain, and a

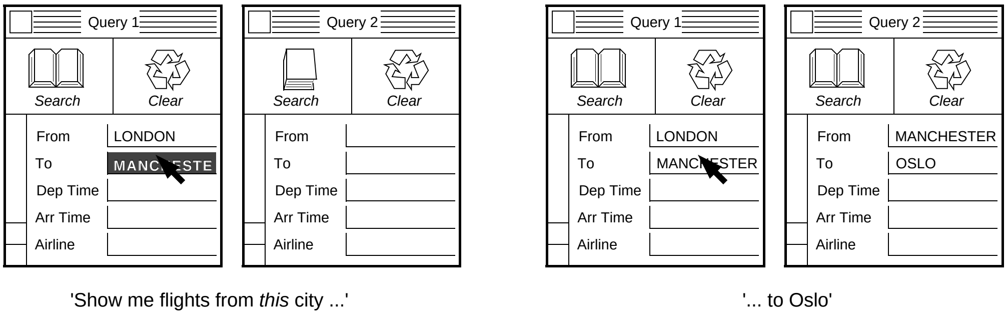

Fig. 1. Deictic blending of speech and gesture in MATIS

presentation that describes the perceivable components of that state. As interactors are only a framework for a specification, they can be used with a variety of mathematical techniques for modelling behaviour. Invariants (properties of the state) and dynamic behaviour, i.e. the evolution of the system through its state space, are described here using MAL. This, rather than the operation notation of Z or VDM, has been used as the axioms required for these examples can be stated and documented concisely.

In the case of MATIS, we are interested in how a user might carry out the task of constructing a query using a combination of speech and gesture. Consequently the state of the model encompasses the contents of the data fields on the form, and the data from the input devices used by the system. We assume that there exists a given type (set of values) called 'name', representing the names of fields on the forms on the MATIS interface. A second given type, 'data', is similarly introduced to represent the set of values that might be provided by the user, either by speech or by pointing.

To model the fusion of information from separate data streams we will represent both speech and mouse data as a sequence of values. For speech, this sequence will contain pairs, each consisting of a database field name and an optional data value. A 'missing' data value (represented by the symbol 'nil') in the speech stream will indicate that the user has employed *deictic* reference; the data for that field will be provided on the mouse stream. The mouse data stream itself is just the sequence of values that have been selected. So for example, if the user utters the query "Flights from *this city* in the morning to *this city*", while using the mouse to select the values 'London' and 'Paris' on the display, the corresponding data streams might look like:

$$\begin{aligned}
\text{speech} &= \langle (\text{From}, \text{nil}), (\text{Time}, \text{morning}), (\text{To}, \text{nil}) \rangle \\
\text{mouse} &= \langle \text{London}, \text{Paris} \rangle
\end{aligned}$$

More generally, we define the type 'value' to be the union of 'data' and the constant 'nil'. A 'slot' (on a form, or on the speech input stream) then is a pair consisting of the field name and a corresponding value.

$$\mathrm{value} \mathrel{\widehat{=}} \mathrm{data} \cup \{\mathrm{nil}\}$$

$$\mathrm{data} \mathrel{\widehat{=}} \mathrm{name} \times \mathrm{value}$$

It is convenient for our present purposes to specify the system with a single interactor, the state of which consists of five components, or attributes. These represent

- the content of each query form;
- the identity of the query form that the user is constructing;
- the sequence of input received along both speech and mouse data streams; and
- the query form that would result from the 'fusion' of the two input data streams.

interactor MATIS

attributes

$\boxed{\textsf{vis}}$	fields	:	qnr × name ⇸ value	- contents of query forms
$\boxed{\textsf{vis}}$	current	:	qnr	- current query
	mouse	:	seq data	- data stream from mouse
	speech	:	seq slot	- data (and holes) from speech
	result	:	name ⇸ data	- outcome of resolving deixis

The annotation '$\boxed{\textsf{vis}}$' is used to indicate that a particular observable is part of the visual presentation of the system. In this case, both the chosen user and the enabled buttons are (potentially) perceivable by the user of the system. The annotation indicates that when these components are perceivable, this is via the 'visual' modality. Observables in the presentation are called 'percepts' [18]. However, just because a percept is defined in the state doesn't mean that it is always perceivable; the conditions under which a percept *could* be perceived are included in the axioms of the system.

The dynamic behaviour of the MATIS system is described in terms of a number of actions. Four actions are defined on the MATIS interactor. The first two actions relate to use of the speech and mouse modalities, and as indicated by the annotations, will be effected by the articulatory and limb channels of the user. The third action, 'fuse' is used to define the effect of performing fusion on the data streams. We will not discuss when fusion should be carried out, or how the results are inserted into the surrounding application by the 'fill' action.

actions

$\boxed{\textbf{art}}$	speak	: name × value	- articulate a data value
$\boxed{\textbf{lim}}$	select	: data	- select a data value
	fuse		- fuse input streams
	fill		- fill in slots on a query form

The remaining part of the interactor is the collection of axioms that inter-relate the observables of the system. MAL extends the usual connectives and quantifiers of first order logic with a modal operator [A] for each action 'A', and two deontic operators that can be used to express that an action is either permitted or obliged under particular conditions. The meaning of each axiom is explained in the accompanying commentary. Axiom 1 defines the effect of the 'speak' action on the speech data stream.

axioms

1 $\quad \mathrm{speech} = \mathrm{X} \Rightarrow [\mathrm{speak}(\mathrm{nm}, \mathrm{d})]\ \mathrm{speech} = \mathrm{X} \frown \langle(\mathrm{nm}, \mathrm{d})\rangle$

If the speech stream holds X, then speaking a name/data pair results in a speech stream with that pair appended to X.

Axiom 2 defines similar behaviour for the 'select' action, though here the new value is a data item that is appended to the stream of mouse input.

2 $\quad \mathrm{mouse} = \mathrm{M} \Rightarrow [\mathrm{select}(\mathrm{d})]\ \mathrm{mouse} = \mathrm{M} \frown \langle \mathrm{d} \rangle$

If the mouse stream holds M, then selecting a data item d results in a stream in which d is appended to M.

The third axiom is an invariant. It requires that all the data in the current query is available in the presentation of the interactor. When expressing properties of percepts, we enclose attributes in boxes to indicate that it is the perceivable representation of the value, rather than the value itself, that is being referred to.

3 $\quad \forall\, \mathrm{n} : \mathrm{name} \bullet (\mathrm{current}, \mathrm{n}) \in \mathrm{dom\,fields} \Rightarrow \boxed{\mathrm{fields(current,n)}}\ \mathrm{in}\ \boxed{\mathrm{MATIS}}$

For any field name 'n', if there is an entry in the fields of the form labelled 'current' for n, then that data is part of the presentation of MATIS.

The development of a model like this can be a useful source of insight in development; by encouraging a developer to document the structure and behaviour of an interface explicitly, latent questions and ungrounded assumptions can be teased out [6,10]. What the model does not (and cannot) address is how the information provided by the system can or should be understood by users, and how users' perception of the system will mediate execution of the tasks for which the system was designed. For example, will users be able to utilise the deixsis capability? As interactive systems make increasingly rich use of different modalities, and rely more on users' often latent knowledge of the world [3], these questions are increasingly beyond the capability of any one modelling approach. To answer questions about the usability of the system captured in the specification, we will need to work within a framework that can make authorative statements about human capabilities and limitations.

2.2 Interacting Cognitive Subsystems (ICS)

ICS [3] is a comprehensive model of human information processing that describes cognition in terms of a collection of subsystems that operate on specific mental codes. Although specialised to deal with specific codes, all subsystems have a common architecture, shown in Figure 2. Incoming data streams arrive at an input array, from which they are copied into an image record representing an unbounded episodic store of all data received by that subsystem. In parallel with the basic copy process, each subsystem also contains transformation processes that convert incoming data into certain other mental codes. This output is passed through a data network to other subsystems. If the incoming data stream is incomplete or unstable, a process can augment it by accessing or buffering the data stream via the image record. However, only one transformation in a given processing configuration can be buffered at any moment. Unstable data streams arise when a transformation cannot 'lock on to' a fixed output for a given input stimulus. This can occur when it is operating on input data containing representations that it has no prior experience of. Coherent data streams (see [3]) may be blended at the input array of a subsystem, with the result that a process can 'engage' and transform data streams derived from multiple input sources. An account of the rationale and evidence supporting this architecture is well beyond the scope of this paper, and involves results from clinical and experimental psychology. The interested reader is referred to [1] and [29] for the background to the ICS model.

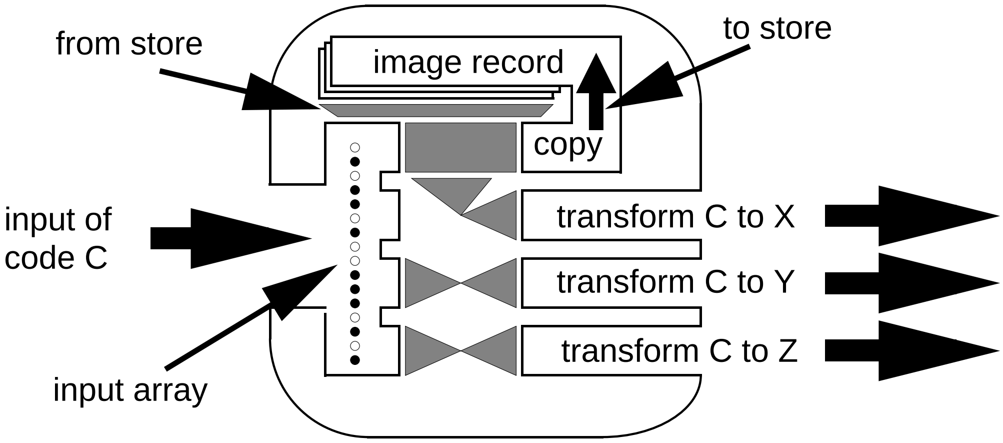

Fig. 2. Generic ICS Subsystem

ICS assumes the existence of 9 distinct subsystems, each based on the above architecture:

Sensory subsystems

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

Structural subsystems

OBJ	object: mental imagery, shapes, etc.
MPL	morphonolexical: words, lexical forms

Meaning subsystems

PROP	propositional: semantic relations
IMPLIC	implicational: holistic meaning

Effector subsystems

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

Overall behaviour of the cognitive system is constrained by the possible transformations and by several principles of processing. Visual information for instance cannot be translated directly into propositional code, but must be processed via the object system that addresses spatial structure. Although in principle all processes are continuously trying to generate code, only some of the processes will generate stable output that is relevant to a given task. This collection of processes is called a *configuration*. The thick lines in Figure 3 show the configuration of resources deployed while trying to direct a hand-controlled input device (such as a mouse) to an icon on a display.

The thick lines in Figure 3 show the configuration of resources deployed while trying to direct a hand-controlled input device (such as a mouse) to an icon on a display.

In order to locate the icon, information arriving at the visual system (1) will be transformed into object code (2) that contains the basic organisation of visual elements on the display. This transformation is written as ∗vis-obj**:** where the ∗indicates information being exchanged with the external world (i.e., arriving from the senses), and the **:** indicates information being exchanged internally. At the same time, the propositional subsystem is copying information about the desired target (3) to its image record, and using :prop-obj: to produce an object code representation (4). When this representation can be blended at the OBJ subsystem with the incoming representation from ∗vis-obj**:**, :obj-prop: will be able to return a matching representation (5) to the propositional subsystem to indicate that a possi-

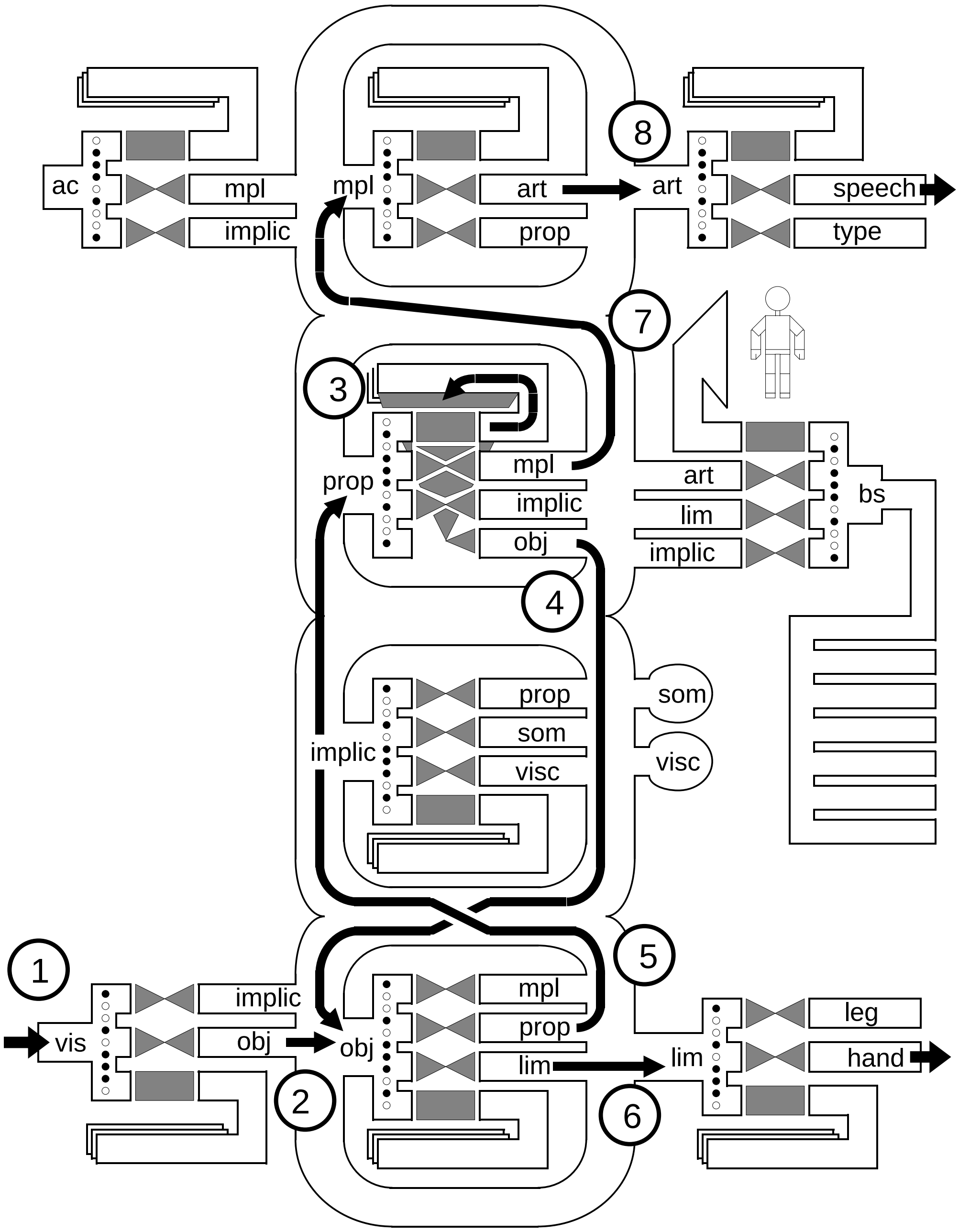

Fig. 3. ICS Resources

ble target has been found. Finally motion of the mouse via the hand is controlled by the limb subsystem through :obj-lim: (6) and then out via **:**lim-hand∗.

While this configuration is actively locating an object, a second sequence of processes could be engaged in producing spoken output, such as "now where is that icon?" This would require the :prop-mpl: process (7) to produce a morphonolexical structure to drive the generation of speech via :mpl-art: and **:**art-speech∗ processes (8).

The key hypothesis 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 a computer-based interface, allowing the models to be integrated directly. As we have used the interactor framework to model devices and interfaces, it has been convenient to use the same structure and to represent aspects of the cognitive model. Once cognitive and device models are in the same form, we can use the expressive power of the formal representation to describe and reason about how the cognitive systems of a user are deployed in performing tasks with the system. In principle, there is no need to use any one specific technique for modelling ICS; the schema notation of Z, PVS theories, or indeed, as Bowman and Faconti [9] have shown, the concept of process could all be used. However, in a recent paper [4], the concept of an interactor has been shown to have value as a generic building block for capturing the behaviour of structured systems, and as always there are trade-offs in how explicitly the structures in such a model are mapped onto the constructs of a specification language. We will return to this issue in the final section of this paper.

The model of ICS we have developed to date is based around the main resources described above - transformation processes and mental representations. A full discussion of our model can be found elsewhere [14,15]; here we provide an introduction using features relevant to the analysis of MATIS.

To represent the concepts of subsystem and representation, we introduce two ‘given’ types, and a relationship over representations; we write ‘p $\approx$ q’ to express the property that representations ‘p’ and ‘q’ are coherent. Critically, we do not address the internal structure of mental representations, for reasons that we will discuss later. Consequently, properties of the relationship, for example consistency and symmetry, are introduced in the full model as axioms.

sys - ICS subsystems, e.g. vis, prop, obj etc.

repr - Mental representations

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

Transformation between mental codes, for example :obj-prop:, are introduced next. Any transformation can be identified by the source and destination subsystems, and for convenience the type ‘tr’ of transformations is modelled as an ordered pair; it would be more correct to define the transformations by an enumeration, but the overhead does not (at the moment) justify the effort. Two functions are introduced to extract the first (src) and second (dst) components of a transformation.

$\mathrm{tr} \mathrel{\widehat{=}} \mathrm{sys} \times \mathrm{sys}$ - names of transformation processes

$\mathrm{src}, \mathrm{dst} : \mathrm{tr} \longrightarrow \mathrm{sys}$

Each transformation process within the ICS subsystems operates on and generates a stream of representations. In most cases, these streams are carried by the internal data network of the architecture, but clearly if cognition is to be located in an environment then it must be possible for streams to both originate (perception) and terminate (action) in the outside world. For convenience, we will write transformations as :src-dst:, ∗src-dst**:** or **:**src-dst∗, depending on whether the stream originates or terminates in the external world (∗tr**:**, **:**tr∗) or is completely internal to the data network (:tr:) of the architecture. That is, ‘∗’ denotes the external world and ‘:’ is the data network. In the remainder of the paper, we will use the term ‘stream’ to refer to the input or output of a transformation, and will use the name of the transformation involved, for example ‘:vis-obj:’ when referring to a specific stream.

A set of transformations involved in some information processing task is called a configuration, while the network of transformations involved in information processing for a particular task is called a flow. For example, Figure 3 shows a flow containing (amongst others) the following chain of transformations:

$$\langle *\text{vis-obj}\mathbf{:}, :\text{obj-prop}:, :\text{prop-obj}:, :\text{obj-lim}:, \mathbf{:}\text{lim-hand}*\rangle$$

The corresponding configuration includes the set of transformations that appear in this sequence. In general, a flow consists of a subset of the transformations that make up a configuration, and a given transformation may occur more than once. Formally, we define the type ‘Config’ to be a set transformations, and the type ‘Flow’ to be a sequence of transformations.

$$
\begin{array}{ll}
\text{Config} & \widehat{=}\ \mathbb{P}\,\text{tr} \\
\text{Flow} & \widehat{=}\ \text{seq}\,\text{tr}
\end{array}
$$

The state of the ICS interactor captures the flows of information involved in processing activities, and the properties of specific transformations 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 function ‘input’ maps each transformation to the representation that is currently available to it as input. As we will see, this input representation may be derived by blending the output of several other processes.

interactor ICS

attributes

$$
\begin{array}{lll}
\text{sources} & : & \text{tr} \longrightarrow \mathbb{P}\,\text{tr} \\
\text{stable} & : & \mathbb{P}\,\text{tr} \\
\text{input} & : & \text{tr} \longrightarrow \text{repr}
\end{array}
$$

The representations being generated by a transformation are given by the relation ‘_ on _’, where ‘p on t’ means that representation p is available as the output of t. All representations arriving at a subsystem are copied to the image record, and the contents of these records are represented by the attribute ‘_@_’ where ‘p@s’ means that representation ‘p’ is part of the image record of subsystem ‘s’.

$$\begin{aligned} \_\ \text{on}\ \_ &: \text{repr} \leftrightarrow \text{tr} \\ \_@\_ &: \text{repr} \leftrightarrow \text{sys} \end{aligned}$$

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 used as input ”downstream”. 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 and this process is identified by the attribute ‘buffered’.

$$\begin{aligned} \text{coherent} &: \text{tr} \times \text{tr} \rightarrow \mathbb{B} \\ \text{buffered} &: \text{tr} \end{aligned}$$

The configuration itself is defined to be those processes whose output is stable and which are contributing to the current processing activity. This processing activity, in turn, consists of a set of flows carrying data through the architecture, and these are represented by the attribute called ‘flows’.

$$\begin{aligned} \text{config} &: \text{Config} \\ \text{flows} &: \mathbb{P}\,\text{Flow} \end{aligned}$$

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

$$\begin{aligned} \text{engage} &: \text{tr} \times \text{tr} \\ \text{disengage} &: \text{tr} \times \text{tr} \\ \text{buffer} & \\ \text{trans} & \end{aligned}$$

The principles of information processing embodied by ICS are expressed as axioms over the model defined above. Axiom 1 defines coherence of data streams in terms of coherence of the representations available on those streams.

axioms

1 $\mathrm{coherent}(t_1, t_2) \Leftrightarrow \mathrm{dest}(t_1) = \mathrm{dest}(t_2) \wedge \forall p, q : \mathrm{repr} \bullet p \text{ on } t_1 \wedge q \text{ on } t_2 \Rightarrow p \approx q$

The output streams from transformations t_1 and t_2 are coherent if and only if they have the same destination, and for any representation p available on t_1, and q on stream t_2, p and q are coherent.

The second axiom defines the concept of a stream's stability. This requires that the inputs to the transformation generating the stream are at least stable. However, coherent input doesn't guarantee stable output, as the input may only be a partial representation of the data that the process needs to generate output. If the input is unstable, then the process will need to be buffered. A configuration is then the set of processes that are generating output that is both stable and which is used elsewhere in the overall processing cycle.

2 $t \in \mathrm{stable}$

$\Leftrightarrow \forall s_1, s_2 : \mathrm{sources}(t) \bullet \mathrm{coherent}(s_1, s_2) \wedge (t = \mathrm{buffered} \vee \mathrm{sources}(t) \subseteq \mathrm{stable})$

A transformation 't' is stable if and only if every pair of streams on which it operates are coherent, and either the transformation is buffered, or the input streams are themselves stable.

3 $t \in \mathrm{config} \Leftrightarrow (t \in \mathrm{stable} \wedge \mathrm{src}(t) \not\in \{\mathrm{art}, \mathrm{lim}\} \Rightarrow \exists s : \mathrm{tr} \bullet t \in \mathrm{sources}(s))$

A stream or process t is part of the processing configuration if and only if it is stable and, unless it is part of an effector subsystem, there is some other transformation 's' that is using the stream from t.

Axioms 4 and 5 concern flows. Any transformation that is part of a flow must be part of the configuration, and similarly if a transformation is in the configuration it must be part of some flow. This is expressed by axiom 4. Axiom 5 captures the 'chaining' property of flows. If two transformations are adjacent in a flow, then the first transformation must be one of the sources used by the second transformation. The symbol '$\frown$' is sequence concatenation.

4 $t \in \mathrm{config} \Leftrightarrow (\exists f : \mathrm{flows} \bullet t \in \mathrm{ran}\, f)$

A transformation 't' is in the configuration if and only if there exists some flow 'f' that contains t.

5 $\forall s, u : \mathrm{Flow};\ t_1, t_2 : \mathrm{tr} \bullet s \frown \langle t_1, t_2 \rangle \frown u \in \mathrm{flows} \Leftrightarrow t_1 \in \mathrm{sources}(t_2)$

For arbitrary flows 's' and 'u', and transformations 't_1' and 't_2', there is a flow in the system containing t_1 followed by t_2 if and only if t_1 is a source of t_2.

Axioms that address unstable data streams, operation of the buffer, and information transfer, have been set out in other papers [14,15], but are not of concern here. Instead, we now show how this formal description of ICS can be used in conjunction with the device model of MATIS.

2.3 A syndetic model of MATIS

MATIS includes the capability of the system to handle deictic input. Deixis is a feature of human-human interaction, so one can make an informal case that it represents a potentially useful tool for human-computer interaction. To explore whether this is in fact the case, we construct a syndetic model by combining the MATIS specification with the model of ICS, allowing us then to conjecture about the conditions under which deixis will be possible to a user of the system.

The MATIS and ICS specifications were developed independently of each other, and in bringing them together into a syndetic model we extend the original models with additional observables (in this case an action) that captures the interplay between the two components. For MATIS, we posit a 'read' action that allows the user to locate some lexical item, such as the name of a city, on the presentation. In a more substantial system model, this action would be bound to the contents of the query forms (see Figure 1) that were available at any time on the screen. This degree of detail however is not essential for illustrating the role played by syndesis in understanding deixis.

interactor MATIS-User

MATIS - include the MATIS spec

ICS - and the ICS framework

actions

read : data - observe the MATIS presentation

The conjoint behaviour of the two agents is captured by three axioms that span the two sets of observables, in MATIS and ICS. The first axiom defines the condition under which it is possible for the user to read an item of data from the presentation. On the system side, there must exist a field on a query such that the value of the field is the data item. On the user side, the configuration must include a data flow from the visual system, through the object and morphonolexical levels, to the propositional subsystem.

axioms

1 $\textbf{per}(\text{read}(\text{d})) \Rightarrow \boxed{\text{d}}\ \text{in}\ \boxed{\text{MATIS}} \wedge \langle *\text{vis-obj}\textbf{:}, :\text{obj-mpl}:, :\text{mpl-prop}:\rangle \in \text{flows}$

It is possible to read some data item 'd' if d is part of a field of a query in the display and the cognitive configuration enables reading.

In this scenario we are concerned with the representations that are being processed within a flow. To capture this idea concisely, we define a relational symbol named 'on-flow' as an abbreviation for a condition involving components of the ICS interactor:

on-flow : repr ↔ Flow

$\widehat{=}$ r on-flow f ⇔ f ∈ flows ∧ ∀ t : ran f • r on t

A representation ‘r’ is on a flow ‘f’ if and only if the flow is part of the processing configuration, and for all transformations that are in the range of the sequence defining the flow, the corresponding representation is available as output of those transformations.

Axioms 2 and 3 address the cognitive requirements associated with the action of selecting a data item with the mouse, and uttering some part of a query. As items on the MATIS display are lexical structures, the mpl and prop systems need to be recruited to find lexical objects (words) on the screen and compare them with the users’ goals. This will require that the representation of the word is on the flow defined by a search configuration suitable for lexicographical data derived from visual input. For speech, the data flow will begin within a PIP loop and then will be processed via the mpl and art subsystems to produce spoken words.

word-search

$\widehat{=}$ ⟨∗vis-obj**:**, :obj-mpl:, :mpl-prop:, :prop-mpl:, :mpl-prop:⟩

speech

$\widehat{=}$ PIP ⁀ ⟨:prop-mpl:, :mpl-art:, **:**art-speech∗⟩

Note that the final three transformations in the ‘word-search’ flow define a processing cycle referred to as a PMP loop. That is, propositional information produced by mpl may be used by processes in prop to construct new mpl representations. The PMP loop and PIP loop needed for speech indicate a cyclic interchange of representations between two or more processes, as described in [14].

2 **per**(select(d)) ⇒ $\boxed{\text{d}}$ in $\boxed{\text{MATIS}}$ ∧ d on-flow **word-search**

If it is possible to select an item (with the mouse) then the item must be part of the display, and a representation of the item must be processed within a flow configured for lexicographical search and comparison.

3 **per**(speak(s)) ⇒ s on-flow **speech**

If it is possible to articulate part of a query then a representation of the phrase must be processed through a data flow that originates as a PIP loop and then results in the production of speech via the mpl system.

Since deixis involves operating on two streams of potentially different representations (one dealing with data to be spoken, the other with data involved in lexical search), we conjectured that there might be a difficulty in using the interface if these streams conflict. We constructed a hypothesis that in order for the user to speak a phrase ‘s’ and select a data item ‘d’ concurrently, the representations of ‘s’ and ‘d’ must be coherent. This can be expressed formally, as the sequent given below.

$$\text{MATIS-User} \wedge \textbf{per}(\text{speak}(s)\&\text{select}(d)) \Rrightarrow s \approx d$$

Details of the proof can be found in [14]. Informally, the result shows that deictic reference, as used in MATIS, places conflicting demands on cognitive resources, a result bourne out by the experience of the developers.

Too often in HCI, the role and value of theory is under-appreciated. It is therefore useful to summarise the process described above. Starting with a model of a specific interface, and a *general* model of a cognitive architecture, we set out to explore a conjecture that the concurrent use of multiple data streams required to achieve deictic reference would place a strong requirement on the user. Reasoning within the syndetic model, we concluded that deictic reference required that the representations being processed on two data streams would need to be coherent; interpreting this result in the context of the models lead us to claim that users would find deictic reference difficult.

It is important to note the role of the mathematics. Describing MATIS and ICS as a set of axioms did not in itself lead to the conjecture about usability, or the subsequent proof. Nor would one expect it to. The use of mathematics here in HCI is no different from its use in any other scientific discipline; it is a tool for representing models of the world and for manipulating those models to test conjectures and carry out calculation. However, this is not to say that the mathematics played no role in discovering the result. The mathematical model makes the role of data streams explicit, and by providing a concise vocabulary for describing the properties and behaviour of these streams, there is a sense in which the formulae *afford* exploration of properties related to stream-based processing. In this sense the mathematical representation enables discovery of these processes in the same way that powerful and expressive bodies of mathematical theory empower physicists to calculate properties of electromagnetic fields or quantum states. Indeed, the successful development of theoretical models to explain and predict the results of experiments was in part due to the existence of mathematics, such as vector spaces, operators, and differential equations in which the observations could be expressed concisely and clearly.

2.4 Achievements

A number of published case studies, among them the MATIS example described here, have demonstrated that by combining user and system models, syndesis allows us to explore and reason about properties of interaction that, we believe, would otherwise remain latent. In the case of MATIS, we claimed that users would find deictic reference difficult, and were able to point to theoretically grounded reasons for *why* this is so. Syndetic models are important here because they make explicit both the chain of reasoning that leads to problem identification, and the fundamental principles or assumptions on which this chain is grounded. In contrast, purely empirical approaches to evaluation can identify that a problem exists, and may localise the context in which it occurs, but without an explicit theory base they lack authority to state the cause of the problem, and consequently do not, in themselves, provide help in identifying solutions.

The idea of bringing together user and system models is independent of the underlying cognitive theory or the representation of that theory. However, we have particular reasons for selecting ICS as the cognitive component, and representing it using a formal description technique. First, ICS has both the breadth of applicability and depth of theory to support the analysis of the kind of novel and sophisticated technology that is moving out of research contexts into social and industrial application. Its scope of application ranges from display structure, through blending of multi-modal data streams, to analysis of clinical depression [3] . This means that ICS brings with it some general results about different types of information processing which can be used within syndetic modelling as specialised forms of the basic theory. There is also evidence in the form of 'expert' systems [24] that significant principles underlying ICS can be represented within a formal framework. As our work draws on the general body of mathematics rather than a restricted deduction apparatus required for an implementation of ICS, we also have some confidence that a comprehensive formal model of ICS is both achievable and tractable. One more point should be mentioned: ICS and FDTs operate at a commensurate level of abstraction. FDTs require the use observables to characterise the intended behaviour of some system; the fact that these systems are usually built in the form of computer software or hardware is irrelevant. ICS also defines a system; one that involves cognitive resources, information flow, and transformation. Thus both user and system structures impose constraints on the processing of information within the overall system. A more operational model of cognition, such as GOMS or SOAR would, in this context, impose a level of operational detail that would make a syndetic model either intractable or too costly for the insight that it generates.

The questions that can be answered by syndetic models of the kind described here (or by any model in fact) are limited by what is represented in the model. A key limitation of the *formalised* ICS model is that representations are treated as abstract, uninterpreted, entities. Properties of representations are addressed only through axioms (see [14]), which on the one hand raises questions about the correctness, completeness and consistency of the axiom set as a representation of the psychological theory, and on the other limits the depth of explanation that can be acquired from a proof. Insight ceases as the level of the axioms, one cannot enquire what lies behind the axioms and why they are as they are. We have not, for example, considered how processes access the contents of their system's image record when in buffered mode. Nor, in the context of the MATIS example, have we described why two representations might or might not be coherent. This is a significant issue, as ICS does allow *different* representations to be used as input to one process, provided these can be blended. Enriching the model to accommodate some aspects of memory access or blending is not difficult in principle, but rather

[3] This is not of theoretical interest; modern user interfaces increasingly make use of affective properties of human cognition [3], and ICS is one of the few (if not the only) comprehensive model of human information processing that incorporates affect and its impact on other levels of cognitive information processing.

more work on developing our understanding of the theory will be required if we are to find the appropriate level of abstraction and framework for capturing these details in a tractable form.

Abstracting away unnecessary detail and clutter is a very important part of any modelling exercise, and for some applications the level of abstraction presented here will be entirely appropriate. Finding a tractable representation for reasoning about interactive systems is only half of the problem. The use of formal methods in conjunction with ICS opens up interesting opportunities to understand better some of the basic science underlying human information processing. However, in order to answer more detailed questions, for example concerning stability, coherence and blending of input streams, it will be necessary to have some theoretical machinery to capture representations. For example, [23] describes a model in which representations consist of basic units of information organised into superordinate structures. It is known that the coherence of representations depends on several issues, including the timing of representations. Whilst it is possible to account for some aspects of the concept of coherence by defining a relation over representations (in the way done here), a more detailed account requires consideration of issues such as timing. How, or indeed, if, it is possible to do this is a question to which we return in the next section.

The behaviour of the MATIS system has been captured at a particular level of granularity. ICS deals with multiple levels of granularity from the existence of flows and particular configurations, down to the specific details of representations and transformation processes. There are modelling techniques in computer science that could be applied to finer levels of modelling; the duration calculus and time-based process calcui (as used, for example, by Bowman and Faconti[9]). However there are two difficulties with taking this route. First, the modelling techniques require detail that either the cognitive science cannot yet supply, or that is regarded as unimportant from a cognitive perspective. Second, as the mathematical language becomes richer, our expressions can require more explicit reference to the mathematical structure. For example, to express the requirement in the duration calculus that some state S persists for at least time τ, one writes:

$$\lceil \mathrm{S} \rceil ;\ \lceil \neg\, \mathrm{S} \rceil ;\ \lceil \mathrm{S} \rceil \Rightarrow \ell \geq \tau$$

Although formally elegant, this formula implicitly encapsulates a mathematical paradigm for working with durations: ”clamping” a duration by capturing the point where it starts and ends, and understanding that the durations before and after the state of interest can become indefinitely small. This level of understanding is not generally of interest to a cognitive psychologist. Of course, validation techniques can be used, e.g. generating examples and counter-examples for axioms, but this is counter productive. It limits the involvement of the psychologists in the process of making the specification, and it defeats in part the rationale for having the formalism in the first place, to serve as a vehicle for having a shared, unambiguous representation of the theory. Duration calculus has been used just to illustrate the point — similar problems can be found in all notations that the authors are aware of. Modal Action Logic, as used in this paper, is no exception. The challenge is to find mathematical languages that match the levels at which the scientists are working and to match these levels within the system model.

This work is also raising more general questions about how we can combine models from different scientific disciplines. This is also a question to which we return in the next section.

3 Prospects

The approach that has been developed so far enables us to reason about user interface issues that depend on the deployment of cognitive resources. This is illustrated in a case study of gestural interaction [13,15] and the analysis of properties of the deixis configurations considered by Bowman and Faconti [9]. The work has focused on the data flows between the resources required to peform some information processing task, and on properties of such flows, for example stability and coherence. The work acknowledges that the flows carry 'mental representations', but the structure and properties of such representations have not been considered. There are two reasons for this: firstly that the cognitive theory underlying representations is unclear (at least from the point of view of modelling), and second, there is a lack of a mathematical framework in which representations can be captured. We may also remark that it is unclear exactly what it is about representations that needs to be captured.

3.1 The dynamics of ICS

Dynamic aspects of ICS include:

- the overall configuration and stability of data streams that exist in the system from moment to moment;
- the mental representations available as input to the processes of a subsystem, and
- the process (if any) operating in extended mode, i.e. that is able to draw on the contents of its image record.

Although the image records of the subsystems can also be seen as a resource, they are somewhat more problematic. Image records contain an episodic record of all input arriving at a subsystem, and through a principal of revival, incoming data at the input array may lead to a process operating on a revived representation from memory. This raises a number of issues.

- What representations are available in memory?
- The different kinds of representations used in tasks (see [2]).
- How the product of revival is related to the content of memory.

The operation and role of memory in cognition has been the focus of considerable work in the applied psychology conmmunity for many years, and the model adopted in ICS builds on these results. However, the scale and complexity of these theories is such they have not been incorporated into the model to date. A key problem is that the operation of memory is linked intimately to the structure of mental representations and the actions of the transducers within the overall system.

There are results available concerning the real-time behaviour of the cognitive architecture. Whilst we have not yet incorporated such results in our modelling approach, Bowman and Faconti [9] have taken some steps in this direction.

Independent of the difficulties involved in developing a (more) comprehensive mathematical model of ICS is the problem of dealing with multiple levels of detail within scenarios. As an example, the architecture allows certain processes to operate in a cycle, with one process producing a data stream that is used by the second to produce a data stream that is again used by the first. Figure 3 illustrates such a situation, with a reciprocal loop involving :obj-prop: and :prop-obj:. The presence of loops such as this is an important consequence of the basic theory and arises in scenarios when, for example, transducers are unable to operate on the available data streams due to a lack of proceduralised knowledge. However, dealing with this kind of behaviour within a state-action model is problematic. The possiblity of interference effects from other data streams entering a subsystem involved in a reciprocal exchange, means, on the one hand, that each processing step within such a cycle should be explicit. On the other hand, the number of cycles is unbounded, and consequently it becomes extremely difficult to describe or reason adequately about scenarios involving this kind of behaviour. This is particularly so in the modal action logic used to this point, as the states mentioned in predicates dealing with evolving behaviour must be specified explicitly in terms of the actions and assumptions through which the state is constructed. The use of richer modal operators might allow some progress to be made in this area, but we are increasingly of the opinion that a different framework, in which it is possible to describe behaviour at variable levels of temporal granularity, is really needed. Approaches to modelling industrial control systems may find application here.

3.2 Mental representations

As noted already, the formal model of ICS acknowledges the existence of mental representations carried on flows, but does not consider the structure or properties of such representations. There are two reasons for this. Firstly the actual cognitive theory underlying representations is still unclear to us, at least when viewed from a formal modelling perspective, and second we lack a mathematical framework in which to capture what can be asserted about representations, at least in a way that is tractable and could support reasoning involving representations.

Leaving aside the general problem of modelling human information processing, there is a considerable body of existing and evolving knowledge about modelling specific aspects of the overall system. The task of modelling human vision has been explored in some depth, for example. The interplay between models of human vision and the mathematical techniques developed for carrying out computer-based image analysis has been particularly rich. The visual subsystem is much more accessible as a subject for experimental investigation than, for example, deeper propositional models of the world.

Sophisticated mdoels of audition and auditory scene analysis have also been set out, again with strong links to practical application in signal generation and processing [12]. There seem to be some common factors between the approaches developed in these two areas and those used in the wider context of signal processing. Whilst this might not be unexpected, it is of particular interest in our context since one of the fundamental assumptions underpinning the ICS framework is that the human information processor can be viewed as a distributed collection of subsystems which are each based on a common architecture. While each subsystem receives particular kinds of mental representations, operationally the systems are the same. So, for example, the operations that the visual system carries out on visual representations should correspond at some level to those that the acoustic system carries out on sound. One of our long-term goals could thus be re-stated as the discovery of mathematical models for these fundamental operations. That these operations also reflect processing at other, in particular, higher, levels of mental processing remains to be validated.

Although we do not yet know what form the mathematics should take, there are a few faint glimmers of light in the darkness, in particular:

(i) Interactions between different representations seem to suggest a wave-like model (analogies with constructive and destructive interference, superposition in quantum mechanics), perhaps represented in the frequency domain.

(ii) Some of the operations that might be carried out on a model of representations are reminiscent of the operator calculus used, for example, in quantum mechanics.

(iii) Models of low-level neural behaviour use mathematics not dissimilar to that underlying simple electrical circuit components such as phase-locked oscillators and filters; in particular the frequency domain plays s critical role [21].

There is some evidence that the human visual system operates at a number of frequencies in parallel. Such effects are well-documented in the acoustic layer, for example the ability to focus on one particular sound source. The notion of operating at multiple levels concurrently resonates with the notion of scale spaces in models of human visual processing [30], though this is an idea that cannot be developed further here.

As stated above these are pointers to directions of further enquiry, and may turn out to be totally misguided.

Identifying and developing a theory about the information processor is only half the problem; we also need to understand how to use the theory. For syndetic modelling, this involves knowing how to express properties of interfaces and devices within the same kind of model. So, while it may well be that a wave-like model of representations gives insight into how blending operates within the cognitive system, we then need some way of expressing the interface to a device in equivalent terms. Likewise, if we develop a formal model that captures the psychological theory underlying revival of representations from the image record, we then need to consider how much of the ”contextual knowledge” of the user needs to be represented in order to reason about interaction. It may in fact be that syndetic reasoning in this situation will be most effective working backwards: in order to use an interface to a desired level of skill, what representations will the user need to have? One compromise between the demands for a detailed model that explains the operation of information processing, and more abstract models that are sufficiently tractable to allow reasoning about concrete problems may be to link the models by refinement. Some preliminary work on exploring this possibility has been carried out in the context of programmable user models (PUM) [11]. PUM, however, derives from an AI approach to simulating cognitive performance, and it is significantly simpler to represent within the framework of refinement than is ICS.

3.3 Social context

Firstly we may remark that safety critical systems typically involve human agents as well as computer agents, and once again we see that to be able to reason about the overall properties of the system we need to be able to reason at some level about the human agents in them. Syndetic modelling may not be the answer to this, but there may be pointers here for ways forward in this endeavour, not least that ICS may provide a context in which the properties of human agents that need to be considered can be elucidated.

Secondly we note that increasingly computer systems are used not by single users, but by groups of users. Emergency vehicle dispatch systems are just one example. In order to understand the behaviour of such systems, we need to be able to consider the behaviour of groups of users, not just single users. We are a long way from being able to do this formally within the context of syndetic modelling, but we may note that a long term goal is to evolve a framework within which emergent group behaviour can be considered as well as individual behaviour.

4 Convergence and Conclusions

Syndetic modelling began as an attempt to capture – and then reason about – the resources and constraints that both copmuting devices and human cognitive processing bring to interaction. However, in teasing out properties of the cognitive architecture, syndetic modelling has begun to play a more general role, helping us to crystalise our understanding of human information processing through the medium of mathematical structures. It also reveals where our understanding is incomplete or vague. Thus through trying to develop axioms to characterise multimodal blending, we found we needed to consider both flows and representations. We have been moderately successful in reasoning about the former, but to go beyond positing properties of coherence and stability we need to understand something deeper about representations on flows, even if we are not yet at the stage of fully understanding what that ‘something‘ is.

Cognitive systems, like computer systems, are built from components, and the behaviour of ICS as a system emerges in part from the behaviours of its constituent elements, transformation processes and image records. By opening up levels of detail that have hitherto been treated as ”givens” — representations, the operation of memory, and proceduralisation, for example, and building from these down to the neurological layer —- we believe new insight will be shed on how we utilise information in the external world. This may have significant practical value in areas such as visualisation, ”universal access”, and emerging technologies such as haptic and kinesthetic interfaces. There is also much interesting work to be done to discover whether some of the mathematics that describes interaction between human and device can be generalised to higher levels of interaction, between humans and artefacts in their environment, or between groups.

Acknowledgements

We gratefully acknowledge the contributions to this work of our colleagues Phil Barnard and Jon May, to people with whom we collaborated in the Amodeus-2 project and those with whom we collaborate now in the TACIT project.

References

[1] Barnard, P., *Interacting cognitive subsystems: A psycholinguistic approach to short-term memory*, in: A. Ellis, editor, *Progress in the Psychology of Language, Volume 2* (1985).

[2] Barnard, P. and J. May, *Cognitive modelling for user requirements*, in: P. Byerley, P. Barnard and J. May, editors, *Computers, Communication and Usability: Design Issues, Research and Methods for Integrated Services* (1993).

[3] Barnard, P. and J. May, *Interactions with advanced graphical interfaces and the deployment of latent human knowledge*, in: F. Paternó, editor, *Interactive Systems: Design, Specification and Verification* (1994), pp. 15–49.

[4] Barnard, P., J. May, D. Duke and D. Duce, *Systems, interactions and macrotheory*, Transactions on Computer Human Interaction **7** (2000), pp. 222–262.

[5] Bellotti, A., S. Buckingham Shum, A. MacLean and N. Hammond, *Multidisciplinary modelling in HCI design ... in theory and practice*, in: *Proceedings of CHI'95: Human Factors in Computing Systems* (1996), pp. 146–153.

[6] Bellotti, V., A. Blandford, D. Duke, A. Maclean, J. May and L. Nigay, *Controlling accessibility in computer mediated communications: A systematic analysis of the design space*, Human Computer Interaction **6** (1996), pp. 357–432.

[7] Blandford, A. and D. Duke, *Integrating user and computer system concerns in the design of interactive systems*, Int. J. Human-Computer Studies **46** (1997), pp. 653 – 679.

[8] Blandford, A., M. Harrison and P. Barnard, *Using Interaction Framework to guide the design of interactive systems*, Int. J. Human-Computer Studies **43** (1995), pp. 101–130.

[9] Bowman, H. and G. Faconti, *Analysing cognitive behaviour using LOTOS and Mexitl*, Formal Aspects of Computing **11** (1999), pp. 132–159.

[10] Buckingham Shum, S., A. Blandford, D. Duke, J. Good, J. May, F. Paternó and R. Young, *Multidisciplinary modelling for user-centred system design: An air-traffic control case study*, in: M. Sasse, R. Cunningham and R. Winder, editors, *People and Computers XI: Proceedings of HCI'96*, 1996, pp. 201–219.

[11] Butterworth, R., A. Blandford and D. Duke, *Demonstrating the cognitive plausibility of interactive system specifications*, Formal Aspects of Computing **12** (2000), pp. 237–259.

[12] Cooke, M., "Modelling auditory processing and organisation," Distinguished Dissertations in Computer Science, Cambridge University Press, 1993.

[13] Duke, D., *Reasoning about gestural interaction*, Computer Graphics Forum **14** (1995), pp. C–55–C–66.

[14] Duke, D., P. Barnard, D. Duce and J. May, *Syndetic modelling*, Human Computer Interaction **13** (1998), pp. 337 – 393.

[15] Duke, D. and D. Duce, *The formalization of a cognitive architecture and its application to reasoning about human computer interaction*, Formal Aspects of Computing **11** (1999), pp. 665–689.

[16] Duke, D. and M. Harrison, *Abstract interaction objects*, Computer Graphics Forum **12** (1993), pp. C–25–C–36.

[17] Duke, D. and M. Harrison, *MATIS: a case study in formal specification*, Technical Report SM/WP17, ESPRIT BRA 7040 Amodeus-2 (1994), available at `http://www.mrc-cbu.cam.ac.uk/amodeus/`.

[18] Duke, D. and M. Harrison, *A theory of presentations*, in: M. Naftalin, T. Denvir and M. Bertran, editors, *FME'94: Industrial Benefit of Formal Methods, Volume 873 of Lecture Notes in Computer Science* (1994), pp. 271–299.

[19] Duke, D. and M. Harrison, *Interaction and task requirements*, in: P. Palanque and R. Bastide, editors, *DSV-IS'95: Eurographics Workshop on Design, Specification and Verification of Interactive Systems* (1995), pp. 54–75.

[20] Goldsack, S., *Specifications of an operating system kernel: FOREST and VDM compared*, in: R. Bloomfield, L. Marshall and R. Jones, editors, *VDM'88: VDM-The Way Ahead, Volume 328 of Lecture Notes in Computer Science* (1994), pp. 88 – 100.

[21] Hoppensteadt, F., "An introduction to the mathematics of neurons, modeling in the frequency domain," Cambridge University Press, 1997.

[22] Kent, S., T. Maibaum and W. Quirk, *Formally specifying temporal constraints and error recovery*, in: *Proc. of the IEEE International Workshop on Requirements Engineering* (1993), pp. 208 – 215.

[23] May, J., S. Scott and P. Barnard, *Structuring displays: a psychological guide*, Tutorial notes, European Association for Computer Graphics, Geneva (1995).

[24] May, P., J. Barnard and A. Blandford, *Using structural descriptions of interfaces to automate the modelling of user cognition*, User Modelling and User Adaptive Interfaces **3** (1993), pp. 27–64.

[25] Moher, T. and V. Dirda, *Revising mental models to accomodate expectation failures in human-computer dialogues*, in: P. Palanque and R. Bastide, editors, *DSV-IS'95: Eurographics Workshop on Design, Specification and Verification of Interactive Systems* (1995), pp. 76–92.

[26] Nigay, L., "Conception et modélisation logicielles des systèmes interactifs," Ph.D. thesis, University Joseph Fourier, Grenoble (1994).

[27] Nigay, L. and J. Coutaz, *A generic platform for addressing the multimodal challenge*, in: *Proceedings of CHI'95* (1995), pp. 98–105.

[28] Ryan, M., J. Fiadeiro and T. Maibaum, *Sharing actions and attributes in modal action logic*, in: T. Ito and A. Meyer, editors, *Theoretical Aspects of Computer Science, Volume 526 of Lecture Notes in Computer Science* (1991), pp. 569 – 593.

[29] Teasdale, J. and P. Barnard, "Affect, Cognition and Change: Re-modeling Depressive Thought," Lawrence Erlbaum Associates, 1993.

[30] Ter Romeny, B. and L. Florack, *A multiscale geometric model of human vision*, in: W. Hendee and P. Wells, editors, *The Perception of Visual Information* (1993).