BEHAVIOUR & INFORMATION TECHNOLOGY, 1997, VOL. 16, NO. 3, 161 – 183

A design-oriented framework for modelling the planning and control of multiple task work in secretarial office administration

WALLY SMITH*, BECKY HILL, JOHN LONG and ANDY WHITEFIELD

Ergonomics and HCI Unit, University College London, 26 Bedford Way, London WC1H 0AP

*now at: Centre for HCI design, School of Informatics, City University, Northampton Square, London EC1V 0HB, UK; email walter@soi.city.ac.uk

Abstract. Design-oriented frameworks are a type of Human-Computer Interaction (HCI) discipline knowledge. They are intended to support iterative 'specify-and-implement' design practice, by assisting designers to create models of specific design problems, within a class of design problem. This paper presents a design-oriented framework for a class of HCI design problem, expressed as a the planning and control of multiple task work in secretarial office administration. The planning and control of multiple task work refers generally to how interactive human-computer worksystems specify and select behaviours for performing multiple concurrent tasks. Secretarial office administration is a sub-class of design problem, in which the work supports communications of the organization commissioning the new worksystem. The framework is based on a conception proposed to support an engineering discipline of HCI. The framework conceptualizes the relationship between an interactive worksystem, its domain of work and the effectiveness, or performance, with which work is carried out. The framework was developed from cognitive science and HCI theory and an empirical case-study of an existing secretarial worksystem. The framework expresses the domain of secretarial work as the state transformation of hierarchies of abstract and physical objects, representing communications carried out by the organization. The description of the secretarial worksystem expresses the relationship between abstract processes of planning, controlling, perceiving and executing, and abstract representations of plans and knowledge-of-tasks. Planning heuristics and control rules reflect general properties of the dynamic work domain, such as external interruptions and temporary opportunities. The framework also expresses the relationship between these planning and control structures and performance. In its current form, the framework is incomplete, but illustrates an approach to the development of design-oriented knowledge. Using this type of knowledge, a designer may reason about potential solutions to HCI design problems concerning planning and control behaviours for carrying out multiple task work for secretarial office administration.

1. Introduction

Concern has been expressed in recent years about the gap between design practice and theory in Human-Computer Interaction (HCI). In the 'Kittle House Manifesto', Carroll (1991:1) characterizes the impact of psychological theory on design as follows:

> 'The most sustained, focused, and sophisticated attempts to develop explicit extensions of academic psychology for HCI have had no impact on design practice (Card *et al*. 1983, Polson and Kieras 1985). On the other hand, some of the most seminal and momentous user interface design work of the past 25 years made no explicit use of psychology at all (Engelbart and English 1968, Sutherland 1963)'.

In a chapter from the same volume, Pylyshyn (1991:39), a cognitive scientist, provides an insight into his own experience of failing to bridge the theory-practice gap.

> '. . . each time I left the problem unsolved because sooner or later I was faced with the choice of either abandoning the original problem of interest and instead doing something that seemed to me both trivial and atheoretical, or else doing something that was relevant to some theoretical issue but had little to do with the practical problem at hand.'

This paper presents one response to the challenge of developing HCI theory which bridges the gap to design practice. The approach offers support for design practice by providing designers with the means to construct appropriate models of their specific design problems. That is, models which support reasoning about design problems and potential design solutions.

To create these models, designers are offered design-oriented frameworks for modelling classes of design problem. A design-oriented framework is a form of HCI theory. For a problem class, it defines entities and interrelationships addressed by the models. To be effective, and so avoid Pylyshyn's dilemma, such frameworks must focus on design concerns, while providing substantive theoretical support. In addition, frameworks must be flexible to address actual problems of concern to designers.

This paper presents such a framework for a class of HCI design problem: the planning and control of multiple task work in secretarial office administration (referred to throughout as the HCI-PCMT-SOA framework). To ensure that design concerns are addressed, the HCI-PCMT-SOA framework was initially constructed top-down, starting from a conception of the general class of HCI design problem. In this way, framework development occurs on three levels and in three stages:

stage 1. HCI framework
stage 2. HCI-PCMT framework
stage 3. HCI-PCMT-SOA framework

The stages progressively narrow the problem class expressed. The HCI framework addresses the general HCI design problem. Within this class, the HCI-PCMT framework addresses the design of planning and control behaviours of interactive human-computer worksystems which decide upon and carry out multiple, temporally overlapping, tasks. Through further refinement, the HCI-PCMT-SOA framework addresses the planning and control of multiple task work of secretarial worksystems, which support multiple communications carried out by their organization. At each stage of development, existing concepts and theory were assimilated into the framework, while preserving its essential design focus inherited from superordinate levels.

Having constructed an initial HCI-PCMT-SOA framework, it was then applied to an empirical case-study of a specific secretarial office. The role of the case-study was to support bottom-up adjustments to the framework ensuring its ability to model appropriate behaviours and tasks. The case study does not constitute validation of the framework. Nor can it be claimed to be a generalization of secretarial work. Ongoing work is aimed at achieving such validation and generalization. Neither does the present paper develop applicable methods or tools for HCI practitioners. The frameworks and models proposed here, however, are considered to be a pre-requisite for such method and tool development.

Section 2 of the paper, which follows, suggests how design-oriented frameworks and models might be developed. It also describes the rationale underlying the present work. In Section 3, existing theories and concepts relating to multiple task work, planning and control and secretarial office administration are described. Section 4 presents the HCI-PCMT-SOA framework, applying and extending a conception of HCI (Dowell and Long 1989). Section 5 instantiates the HCI-PCMT-SOA framework in an empirical case-study of secretarial work. Finally, Section 6 offers some conclusions, (readers preferring illustration of the framework's concepts, prior to their abstract descriptions, may wish to read Section 5 before Section 4. Figure 2 provides general orientation).

2. Developing design-oriented frameworks and models for HCI

2.1. *Knowledge to support HCI design practice*

Long and Dowell (1989) propose the discipline of HCI as the application of HCI knowledge, to support design practices, intended to solve HCI design problems. The concern here is the nature of the knowledge which supports the design practice. Two existing forms of HCI knowledge, *craft* and *applied science* (Long and Dowell 1989), have serious shortcomings.[1] First, craft knowledge exists implicitly in the expertise of experienced designers. Thus, it is not publicly available for inspection and development and has an unknown scope of application. Second, applied science knowledge, from relevant academic disciplines (such as Psychology, Linguistics and Sociology), although validated supports design only implicitly and indirectly. The knowledge supports *explanation and prediction*, and so understanding, rather than *diagnosis and prescription*, and so design.

Long and Dowell (1989) describe a third type of knowledge to support HCI practice: engineering principles. Such principles would be validated with respect to their support for design. They would support the design of general solutions to general classes of HCI design problem.

The development of such validated engineering principles represents a long-term goal for an engineering design discipline of HCI, and is beyond the scope of the present work. The intention, here, is to propose a form of HCI knowledge – *design-oriented frameworks* – which is both explicit and supports design directly unlike craft and applied science knowledge. Unlike validated engineering principles, however, frameworks are intended to support design in the near, rather than distant, future. Such frameworks provide the basis for modelling specific design problems. Their purpose is to enable designers to reason more effectively about potential design solutions. Frameworks lack the 'guarantee' of validated engineering principles. Instead they support practices of 'specify-and-implement'. That is, practices where design proceeds through iterations of successive cycles of specification and implementation. Such frameworks support the designer in producing better specifications at an earlier stage of design, thus reducing costly iteration.

Frameworks and models pervade the HCI literature (see Long 1987, Whitefield 1990). However, their support for design is unproven, and the extent of their design orientation is questionable. For HCI knowledge to support design, it must support the expression of HCI design problems (it is, in part, the lack of such support which renders craft and applied science knowledge inadequate). In turn, an expression of HCI design problems must reflect a complete and coherent ontology of HCI. That is, a conception of those entities constituting the scope of the HCI discipline (Long 1991). Dowell and Long (1989) have developed a conception of HCI in which they propose a general, complete and coherent ontology comprising: (i) an *interactive worksystem* – the to-be-designed system comprising users and computers, (ii) a *domain of application* – the work to be carried out by the worksystem, and (iii) *performance* – the effectiveness with which work is carried out.

Validated engineering principles would prescribe, for a given general class of design problem, the class of user behaviours and computer behaviours, constituting the class solution. Design-oriented frameworks and models, in contrast, facilitate reasoning about (rather than applying prescriptive principles to) the relationship between: user and computer behaviours; the work of the domain of application; and performance for the purposes of design.

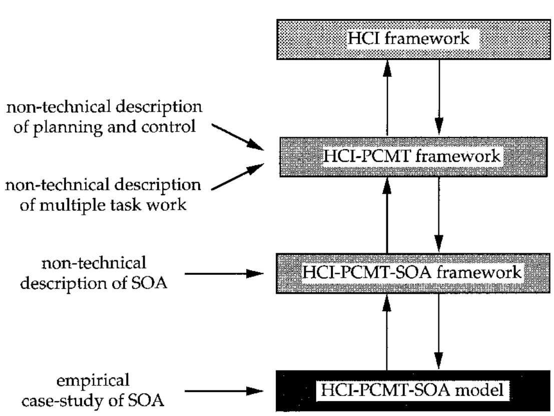

Figure 1. The strategy used to develop a design-oriented Human–Computer Interaction (HCI) framework of the Planning and Control of Multiple Tasks (PCMT) in Secretarial Office Administration (SOA).

To develop such a framework, for the class of HCI-PCMT-50A design problem, successive frameworks were constructed to address increasingly narrow sub-classes of design problem (Figure 1). Finally, a specific secretarial worksystem and domain were modelled. Framework-model development proceeded through iterative adjustments to satisfy two types of constraint. First, were the top-down constraints of Dowell and Long's conception of HCI, which imposed the desired design orientation. Second, were the bottom-up constraints of the empirical observations which ensured potential address of an actual design problem.

Frameworks were developed at three levels (HCI, HCI-PCMT, HCI-PCMT-SOA) to achieve a synthesis of these two types of constraint. In addition, the hierarchical approach facilitated the development of related design-oriented frameworks. For example, on-going work by the present authors concerns the frameworks of HCI-PCMT-MR (medical reception) (Hill *et al.* 1993, 1995) and HCI-PCMT-LSP (legal service provision).

2.2. *Theoretical basis of the HCI-PCMT-SOA framework*

The present framework is 'ecological', in the Gibsonian sense of requiring the inclusion of a domain (here the work) external to the system of concern (here the interactive worksystem). The approach is thus similar to the work of others (e.g., Rasmussen and Vicente 1990). In addition, the to-be-designed worksystem is concerned with the interactive combination of the human and computer, and not simply the computer hardware and software. An instance of this approach is seen in Hutchins' (1995) description and analysis of a flight cockpit, including pilots and technology, as a 'new unit of analysis'. Consistent with a cognitive engineering perspective (Norman 1986), the framework aims to model the cognitive behaviours of a 'joint-cognitive system' (Woods and Hollnagel 1987), considered in relation to their task 'world' (see also Woods and Roth's (1988) 'cognitive systems triad'). Unlike Gibson, and more aligned with cognitivist approaches, the current framework attempts to make explicit the qualitative relationships, between the representations and processes supporting the joint cognitive behaviours of planning and control for multiple task work. Design-oriented models express the relationship between the worksystem and its domain. Such models might be considered to support performance-centred design (in contrast to technology-centred design and user-centred design (e.g., Norman and Draper 1986)).

Developing the framework was also empirical, involving naturalistic observations of secretarial behaviour. Such observations were considered essential, at least in this early phase of construction (Broadbent 1971). Applying the framework to work situations was assisted by the distinction between physical entities and abstract entities. Physical entities are those which are directly observable. Abstract entities are inferred. For example, in the secretarial domain a letter is an (observable) physical object, which is (part of) the physical embodiment of an abstract communication between parties. In the secretarial worksystem, the (observable) physical behaviour of crossing an item off a 'things-to-do-list', is inferred to be the (physical) embodiment of an abstract plan-update behaviour. To infer the abstract correlates of physical behaviours required a fine-grained protocol analysis of secretarial work. The analysis was supported and confirmed by interviewing the secretary. Together, this information was interpreted with reference to the work and organizational contexts. This interpretation of events, hidden to the casual observer, and available only through participants involved in wider contexts, is informally akin to methods of ethnography, as applied to technology use, (e.g., Hughes *et al.* 1993, Suchman 1987). The present concern, however, was not primarily social.

Finally, the ultimate target of the framework is to support the design of interactive human-computer worksystems. The development of the framework here, however, was based on a study of the planning and control behaviours of a non-computerized worksystem. This basis is considered acceptable. Indeed, to address the domain appropriately, it is often desirable to exclude existing technology from empirical investigations, particularly if that technology is at an embryonic stage of development. This point is widely recognized in the field of computer-supported co-operative work (Greenberg 1991), although less so for single user research in HCI.

3. Multiple task work, planning and control, and secretarial office administration

This section provides descriptions of multiple task work, planning and control and secretarial office administration. These descriptions contribute to the development of the framework of HCI-PCMT-SOA (see Section 2.1; Figure 1). They have been drawn from existing theories and related literature. These descriptions are presented below, using some of the terms from Dowell and Long's (1989) conception of HCI. This use facilitates the assimilation of the descriptions into framework development in Section 4.

3.1. *Multiple task work*

Multiple task work requires a user, as part of the interactive worksystem, to perform distinct, but temporally overlapping tasks. Each task potentially competes for worksystem behaviours. Multiple task work represents an important concern for system designers. Performing overlapping tasks is likely to influence the effectiveness with which work is carried out. With increasing development of integrated computerized support, technologies such as broadband communication networks and 'multi-tasking' environments, multiple task work has grown in importance as a design issue.

Experimental psychology research has investigated situations in which people divide their attention between two tasks, or in which they attend selectively to one task in the presence of distracting stimuli (e.g., Cherry 1953, Shiffrin and Schneider 1977). Similar work situations have also been extensively studied by human factors researchers (e.g., Damos 1991). However, the bulk of this research focuses on the specific problems of the near-simultaneous processing of more than one stimulus source or channel. In contrast, the term 'multiple task work' in this paper characterizes situations in which more than one task is carried out concurrently over relatively long and overlapping periods of time. Successful performance of such multiple task work is associated with planning and control behaviours, rather than with abilities of selective and divided attention.

Multiple task work has been discussed by Beishon (1969), who investigated the control by an ovenman of a number of temporally overlapping baking processes. The design implications of a user, switching between various computer-based activities, have also been considered (Bannon *et al.* 1983, Card and Henderson 1987, Cypher 1986, Miyata and Norman 1986). Such research has studied the need to create reminders to resume suspended activities, following interruption, and the provision of contextual cues to support resumption. These empirical observations are useful. However, they are not expressed within a design framework, which would provide support for a class of HCI design problem. For example, research on switching between computer-based activities blurs the distinctions between *tasks* and *activities*. Consequently, little reference is made to the work goals of the system. Consideration of goals would be essential in the design of such systems. In addition research of this kind has made little reference to notions of planning – both how the worksystem specifies its work in advance – and how plans are adapted in response to interruptions and unpredictable events.

To characterize multiple task work requires a single task to be defined. Here, a task is considered to be part of the work carried out in the domain of the worksystem.

A task is thus conceptually distinct from the worksystem itself and its behaviours. A single task for an air traffic controller, for example, might be defined as the safe passage of a single aircraft through a sector of controlled airspace. As air traffic control worksystems typically manage several aircraft at the same time, air traffic control qualifies as multiple task work. A description of secretarial office administration as multiple task work is provided in Section 3.3., including the characterization of a single secretarial task.

3.2. *Planning and control*

In multiple task work, a critical determinant of the human-computer worksystem's effectiveness is how well it manages its own behaviour. That is, how decisions are made about which behaviours to carry out and when to execute them. In this paper, how a worksystem organizes and temporally structures its behaviour is based on: planning; control; and the relationship between planning and control and the worksystem's execution and perception behaviours. Planning, here, is the specification of what, and/or how, tasks are to be accomplished. Control concerns making decisions about what behaviours the worksystem carries out next, at any particular moment (Smith *et al.* 1992).

Planning in cognitive science has been influenced by Newell and Simon's work on problem-solving (Newell *et al.* 1958, Newell and Simon 1972). They showed how a system could generate a sequence of to-be-performed operations to bring about a desired state transformation in a given problem space. For example, Miller *et al.* (1960) state that 'when an organism executes a plan he proceeds through it step by step, completing one part and then moving to the next' (p. 17).

Many researchers have identified the limitations of the concept of plans as complete and fully-elaborated specifications of executable behaviours to achieve a goal state (e.g., Hayes–Roth 1985, Wilensky 1983). Ambros–Ingerson (1986) considered situations in which the planning environment is either too complex to be modelled by the planner, or is dynamic and subject to unpredictable changes. Worksystems faced with such complex and/or dynamic environments would gain little by attempting to construct complete and fully-elaborated plans. Such systems might better utilize partial and/or abstract plans, commence execution, and then accommodate new and unexpected states of the environment, as they occur.

The necessity, in complex and/or dynamic environments, of using plans which do not *completely* specify how an overall goal is to be achieved, requires the worksystem to make control decisions. That is, decisions about what behaviour to carry out next at any particular moment. In such situations, plans serve as resources for guiding control decisions, rather than precise specifications of behaviours to be executed. Such a plan might specify only abstract behaviours, or only some behaviours at an executable level. Alternatively, it might instead specify goals and sub-goals to be achieved independently of the associated behaviours. In all of these cases, planning requires complementary control decisions about how behaviours are to be carried out at their time of execution.

The need to take account of new and unexpected events, as they occur in complex and dynamic environments, places further constraints on the way planning is carried out. The worksystem must be capable of perceiving new states of the environment. Further, it must be capable of terminating or suspending sequences of execution behaviours, when replanning becomes necessary. Thus, control decisions are needed not only for selecting execution behaviours, but also for switching between execution, perception and further planning behaviours.

In the context of HCI, many interactive worksystems carry out work in complex and dynamic domains (e.g., Hollnagel *et al.* 1988). Thus, the classical view from cognitive science of planning as specification of executable behaviours, is unlikely to be appropriate (Young and Simon 1987). The view of partial and/or abstract planning with complementary control decisions would appear more appropriate as a basis for a design-oriented framework. HCI research supports such a view. For example, Suchman's (1987) characterization of 'situated actions' and 'plans-as-resources' with respect to peoples' interaction with devices. Similar support comes from accounts of programming, which have identified the importance of opportunistic behaviours, as well as top-down planning (e.g., Davies 1991). Furthermore, work in system development has also recognized the failures of systems which determine behaviours only by complete and fully-elaborated executable plans. For example, Payton *et al.*'s (1990) account of computerized navigation in automated land vehicles.

3.3. *Secretarial office administration*

The framework described here was developed for the domain of secretarial office administration. This domain was selected because: (1) it is a *prima facie* example of multiple task work; (2) it requires planning and control; and (3) it is a target for the design and development of supporting computerized technology.

Secretarial worksystems can be characterized by the types of behaviour they require. For example: typing;

filing; making appointments; dealing with post; answering telephone inquiries; relaying messages; record-keeping, etc. Similar lists of activities have been classified as secretarial by Engel *et al.* (1979) and Newman (1980).

For present purposes, however, it is necessary to conceptualize secretarial work independently of any particular worksystem or type of behaviour. This point has been made elsewhere in the general context of office system design (e.g., Newman 1980, Schafer 1988).

Previous accounts of office work and technology have been constructed from a variety of perspectives: functional; business; social; and organizational (Hirschheim 1985; Schafer 1988). Simple functional views have conceived office work as the transformation of messages (e.g., Schmidt 1988). Such views have been criticized for failing to incorporate social and organizational factors (e.g., Curran and Mitchell 1982, Hirschheim 1985). Here, a functional view is developed which attempts to incorporate social and organizational factors. Such factors are likely to be relevant to the design of secretarial worksystems.

Every secretarial worksystem is part of a wider system – an organization. For an organization to carry out its work effectively, it must communicate both internally, between different parts of the organization, and externally, between parts of the organization and other organizations. In the framework, secretarial work is conceived as the support of organization communications, both internal and external. The provision of support for organization communications may require a variety of activities such as passing on messages, preparing documents and arranging meetings.

The provision of support for a single communication can be regarded as a single secretarial task. Each communication involves a sender (or group of senders) transmitting information to a receiver (or group of receivers). Secretarial worksystems are typically engaged in the support of several, temporally overlapping, communications at any one time. Secretarial work is, thus, an instance of multiple task work.

4. HCI-PCMT-SOA framework

The previous section outlined informal descriptions of multiple task work, planning and control and secretarial office administration. This section now presents a framework to express the HCI-PCMT-SOA class of design problem. The description of the framework in the paper is complete for this state of its development.

The framework is expressed as a set of axioms and constructed in three stages, as shown in Figure 1. First, the HCI framework (Section 4.1), second, the HCI-PCMT framework (Section 4.2), and last, the

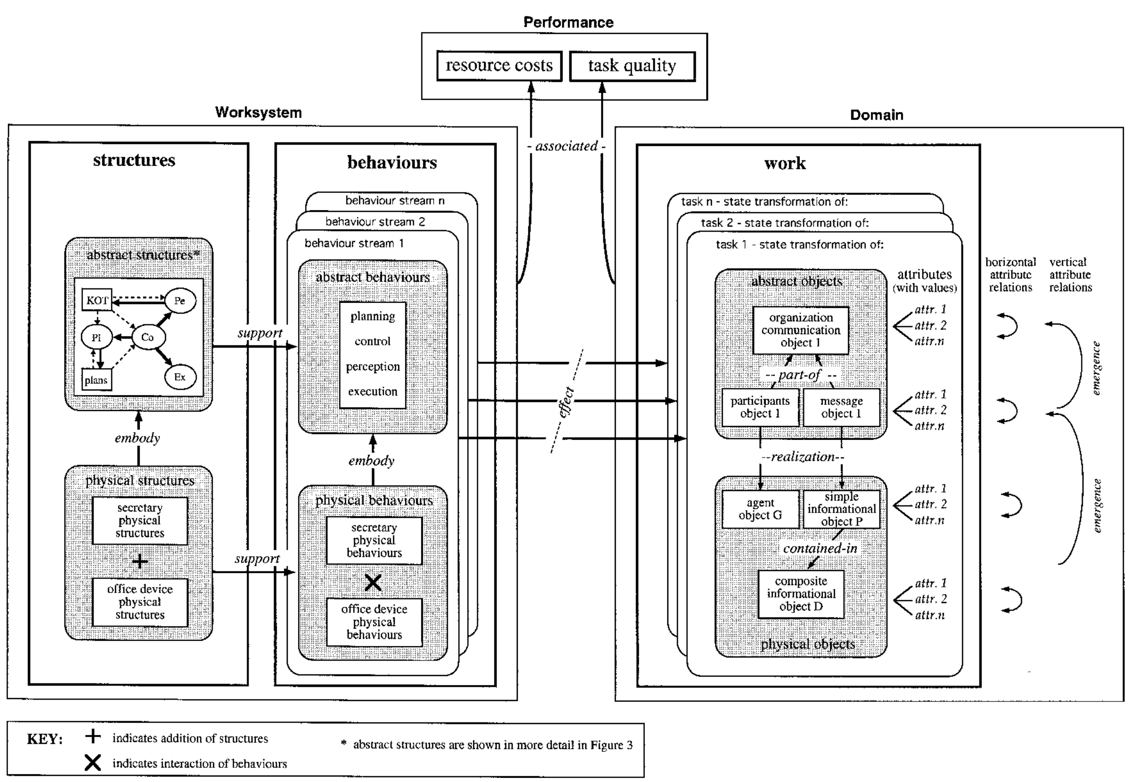

Figure 2. A design-oriented framework of HCI-PCMT-SOA.

HCI-PCMT-SOA framework (Section 4.3). Figure 2 depicts the resulting HCI-PCMT-SOA framework and therefore reflects all stages of the development.

The scope of the framework is the entities and relationships which express HCI design problems and their possible solutions. Similarly, the HCI-PCMT framework and the HCI-PCMT-SOA framework have as their scopes the entities and relationships, which express their respective design problems and possible solutions.

4.1. *Framework Level 1: HCI*

The purpose of the HCI framework is to express design problems to aid a designer to reason about possible design solutions, in a specify-and-implement type of design practice. The axioms for the HCI framework are based on a partial and selective application of Dowell and Long's (1989) conception for HCI. This conception was chosen because of its claimed design orientation, coherence and completeness. Alternative conceptions of HCI have also been proposed (e.g., Carroll and Campbell 1989, Norman and Draper 1986, Storrs 1989, 1994).

Axiom 1.1. HCI design problems: HCI design problems and their solutions, generated by specify-and-implement design practice, entail the specification of the implementable *structures* and *behaviours* of a *worksystem*, comprising *user structures and behaviours* (U) and *device structures and behaviours* (D), which carry out the *work* of its *domain*, such that the *actual level of performance* (P_A) falls within some *desired level of performance* (P_D).

HCI design problems can be expressed as:

specify $\{U \times D\}$

such that $\{U \times D\}$ produces $P_{A \to D}$

where:

U expresses user (structures and) behaviours

D expresses device (structures and) behaviours

P expresses performance

$A \to D$ denotes actual level falling within desired level

$P = fn\ \{Q, K\}$

Q expresses the task quality of the work achieved in the domain

K expresses the resource costs incurred by the worksystem

The following three sections describe, in greater detail, the three central concepts underlying this expression: the domain, the worksystem and performance.

Axioms 1.2. Domain: The domain is conceptualized as those *objects*, whose *state transformations* constitute the *work* carried out by the worksystem.

Domain objects may be *abstract* or *physical* in nature.

Each object has a set of *attributes* with variable *values*. The state of each object is determined by the combined values of its attributes.

Attributes of the same or different objects may be related. That is, an attribute may depend for its value on one or more other attribute values.

Objects may be related by means of their attributes. Particular types of object relationship lead to the domain being structured in particular ways. For example, part-whole relationships, between objects, lead to a domain with a hierarchical structure.

Axioms 1.3. Worksystem: The worksystem is conceptualized as a behavioural system comprising the interacting *user behaviours* (supported by *user structures*) and *device*[2] behaviours (supported by *device structures*). The combined purpose is to achieve a common goal of carrying out the work of the domain.

Worksystem behaviours *effect* the work of the domain. That is, they bring about the required state transformations of the objects of the domain.

The worksystem is conceptualized as both *abstract* and *physical*. Physical structures *embody* abstract structures, and physical behaviours *embody* abstract behaviours.

Abstract structures comprise *representations* and *processes*. Abstract representation structures refer, for example, to the worksystem's knowledge, databases or information stores. Abstract process structures refer, for example, to the worksystem's procedures, methods or heuristics.

Abstract structures *support* worksystem abstract behaviours when abstract process structures, such as procedures, act on abstract representation structures, such as a database. Similarly, worksystem physical structures *support* worksystem physical behaviours. (The latter are not differentiated further in subsequent sections, since the concern here is primarily with the abstract behaviours associated with planning and control.)

All abstract behaviours are embodied in some physical behaviour(s). All physical behaviours embody some abstract behaviour(s).

All abstract structures are embodied in some physical structure(s). All physical structures embody some abstract structure(s).

Axioms 1.4. Performance: Performance is an expression of the effectiveness with which the work of the domain is carried out by the worksystem.

Performance is some function of: (1) the *task quality*, associated with the domain work carried out, and (2) the *resource costs*, associated with worksystem structures and behaviours.

Designers following specify-and-implement design practice need to reason about the *actual level of performance* , i.e, that which is achieved by a worksystem, with respect to some *desired level of performance*. Desired performance characterizes the performance of the worksystem to be designed.

4.2. *Framework Level 2: planning and control of multiple task work*

The purpose of the HCI-PCMT framework is to express the class of HCI-PCMT design problem. A framework of HCI-PCMT is now developed by expanding selected parts of the HCI framework (Section 4.1). This expansion is carried out by adding new axioms. These axioms concern: concepts of worksystem abstract structures and worksystem abstract behaviours, to incorporate concepts of planning and control; and the concept of the work of the domain, to incorporate the concept of multiple task work.

Axiom 2.1. HCI-PCMT design problems: HCI-PCMT design problems and their possible solutions, generated by specify-and-implement design practice, entail the specification of the implementable *planning* (*structures* and) *behaviours* and *control* (*structures* and) *behaviours* of the *user* (U_{PC}) and *devices* (D_{PC}) of the *worksystem*, such that when they interact with the *perception* (*structures* and) *behaviours* and *execution* (*structures* and) *behaviours* of the *user* (U_{PE}) and *devices* (D_{PE}), they carry our *multiple task work* such that the *actual level of performance* (P_{PCMT_A}) falls within some *desired level of performance* (P_{PCMT_D}).

HCI-PCMT design problems can be expressed as:

specify $\{U_{PC} \times D_{PC}\}$
such that $\{U_{PC} \times D_{PC}\} \times \{U_{PE} \times D_{PE}\}$ produces $P_{PCMT_{A \to D}}$

where (following on from Axiom 1.1):
PC denotes planning and control
PE denotes perception and execution
PCMT denotes planning and control of multiple task work
$P_{PCMT} = fn\ \{Q_{MT}, K_{PC}\}$
Q_{MT} expresses the task quality of the multiple task work achieved in the domain
K_{PC} expresses the resource costs associated with planning and control behaviours

Axiom 2.2. HCI-PCMT domain: multiple task work: Relationships between domain objects give rise to different *levels of description*. Abstract objects constitute higher level descriptions of physical objects, and some abstract objects may be higher level descriptions of other abstract objects.

Vertical relationships exist between the values of attributes at different levels of description. Values of attributes at higher levels of description are determined by an *emergence* relationship to the values of attributes at lower levels. *Horizontal relationships* exist between the values of attributes at the same level of description.

A *task* is the required state transformation of a single abstract object at the highest level of description, including all the lower level transformations associated through object relationships.

Multiple task work is that domain work in which, at the highest level of description, there are typically two or more objects undergoing independent, but temporally overlapping work transformations.

A *sub-task* is some part of the state transformation which constitutes a task. It is a sub-transformation (to anticipate an example from secretarial office administration, the task of a supported organization communication, by a posted letter, may comprise the sub-tasks of: the transformation of the letter to 'typed'; the transformation of the envelope to 'prepared'; the transformation of the post-item to 'prepared'; and transformation of the post-item to 'posted'. All sub-tasks require separate associated attribute value changes).

Natural breaks in a task are the temporal boundaries (of whatever duration) between component sub-tasks, when the sub-tasks are effected in sequence.

Temporary *opportunities* may occur in the domain, during which certain transformations of domain objects can be effected by behaviours with relatively low associated resource costs and/or relatively high task quality.

Temporary *interruptions* may occur in the domain, during which certain transformations of domain objects demand particular behaviours from the worksystem to avoid loss of task quality and/or incurring of greater resource costs.

Axiom 2.3. HCI-PCMT worksystem: planning and control behaviours and structures: The temporal sequence of worksystem behaviours may be separated into two or more *behaviour streams*, where each behaviour stream is associated with a single task.

Four types of abstract behaviour are generic to the worksystem and undifferentiated between users and devices. These behaviours are *planning*, *control*, *perception* and *execution*.

The four types of abstract behaviour are supported by abstract structures, also undifferentiated between users and devices. These abstract structures are four types of process, corresponding to the four types of behaviour. That is, a *planning process*, a *controlling process*, a *perceiving process*, and an *executing process*. There are two types of representation: a *plan representation* and a *knowledge-of-tasks representation*. (Figure 2 shows the abstract planning and control structures within the framework. Figure 3 shows the relationship between the types of abstract structure in greater detail[3]. Tables 1 and 2 show examples of the structures which support planning and control behaviours. These are explained more fully in Section 5).

Perception behaviours are those whereby the work-system detects and records the values of domain object attributes. The states of domain objects form the contents of the knowledge-of-tasks representation. Perception behaviours update the contents of the knowledge-of-tasks representation, based on their reading of the domain.

Execution behaviours are those which carry out the work of the worksystem directly by transforming the values of domain object attributes.

Planning behaviours are those which specify what and/or how tasks will be accomplished in terms of required object state transformations and/or required worksystem behaviours. These specifications form the content of the plan representation. Planning behaviours update the contents of the plan representation, based on their reading of the contents of the knowledge-of-tasks representation and the existing contents of the plan representation.

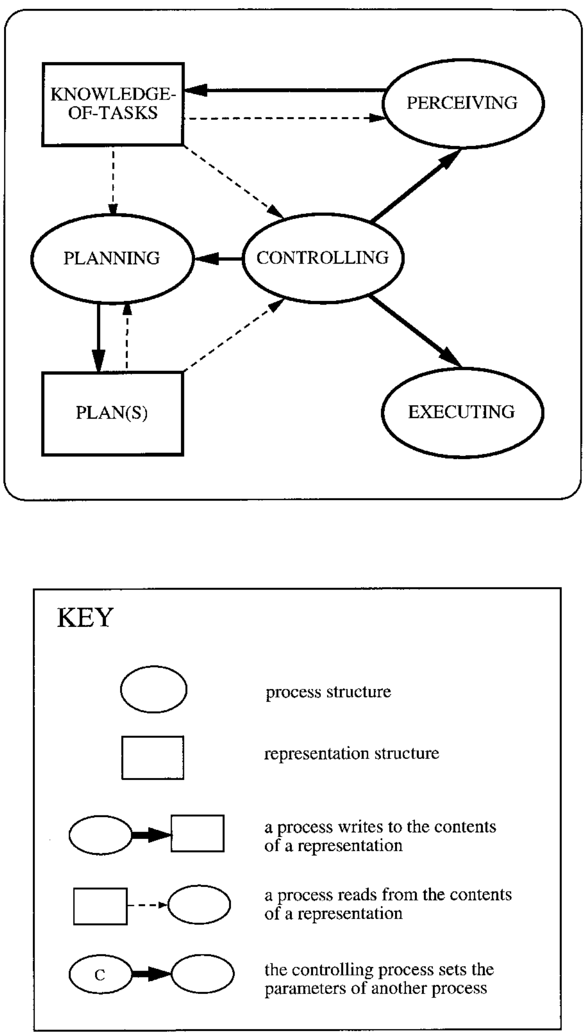

Figure 3. The HCI-PCMT framework: abstract worksystem structures.

Table 1. HCI-PCMT-SOA worksystem model: process control structures supporting control behaviours for multiple task work (expressed as control rules).

control rule 1:	*plan reading at task finish and natural breaks*
condition:	on registering a task finish or a natural break in the current task in knowledge-of-tasks
action:	read the plan to select which task (including the current one) to start/continue/resume the associated behaviour stream.
control rule 2:	*external interruption switching*
condition:	on registering an external interruption in knowledge-of-tasks
action:	finish/stop/suspend the current behaviour stream and start/continue/resume the behaviour stream related to the external interruption.
control rule 3:	*opportunity switching*
condition:	on registering an opportunity in knowledge-of-tasks which relates to one or more tasks in the plan
action:	start/continue/resume behaviour streams (including current task) which carry out the tasks associated with the opportunity as represented in the plan.
control rule 4:	*suspension avoidance*
condition:	on carrying out a finish/stop/suspend behaviour stream, where there is some flexibility over timing
action:	favour finish/stop over suspend, if possible.
control rule 5:	*plan maintenance*
condition:	on finding the current plan insufficiently elaborated and/or scheduled (i.e., below some criterial level)
action:	instruct the planning process to elaborate and/or schedule the plan further until the criterial level is reached.
control rule 6:	*knowledge-of-tasks maintenance*
condition:	on perceiving new information about a task
action:	instruct the perception process to obtain sufficient information (above some criterial level) for planning.

Table 2. HCI-PCMT-SOA worksystem model: process planning structures supporting planning behaviours for multiple task work (expressed as planning heuristics).

planning heuristic 1: representation of opportunities
Make explicit in the plan opportunities which might occur and which relate to task progress.

planning heuristic 2: sharing behaviour across tasks I: grouping of physical objects
Where possible and desirable, merge sub-tasks and tasks together through the planned grouping of physical objects into composite physical objects which can be transformed as a single object, thus progressing all merged tasks simultaneously.

planning heuristic 3: prioritization of suspended tasks
Prioritize suspended tasks (over stopped tasks).

planning heuristic 4: sharing behaviour across tasks II: adjacent scheduling
Schedule sub-tasks adjacently which require the same execution and/or perception behaviours.

There are two types of planning behaviour: *plan elaboration* behaviours, and *plan scheduling* behaviours. Plan elaboration behaviours involve the specification of tasks and/or behaviours in greater detail. Plan scheduling behaviours require the specification of when tasks and/or behaviours are to be carried out.

Control behaviours select which behaviours are to be carried out next at any time. Control behaviours set the *parameters* of the planning, perceiving and executing processes. Thus, they configure the behaviours supported by those processes, based on their reading of the contents of the knowledge-of-tasks representation and the contents of the plan representation.

Control behaviours determine switching, between the different behaviour streams relating to multiple task work, in the following ways. Control behaviours may *start* and *finish* a behaviour stream at the absolute beginning and end of a task transformation, respectively. Control behaviours may *stop* a behaviour stream at a natural break in the task transformation, and later *continue* the behaviour stream from the point at which it has been stopped. Finally, control behaviours may *suspend* a behaviour stream during a sub-task (i.e., not at a natural break) and later *resume* the behaviour stream from the point at which it has been suspended.

Axiom 2.4. HCI-PCMT performance: Performance is some function of: (1) the task quality associated with the multiple task work carried out; and (2) the resource costs associated with the worksystem structures and behaviours of planning and control (incurred by the worksystem as a whole).

4.3. *Framework Level 3: secretarial office administration*

The purpose of the HCI-PCMT-SOA framework (Figure 2) is to express the class of HCI-PCMT-SOA design problem and possible solutions. The framework is now developed by adding axioms to those of the HCI-PCMT framework. The new axioms chiefly concern the particular type of multiple task work involved in secretarial office administration. The worksystem is simply particularized to the secretarial case at both physical and abstract levels.

Axiom 3.1. HCI-PCMT-SOA design problems: HCI-PCMT-SOA design problems and their possible solutions, generated by specify-and-implement design practice, entail the specification of the implementable *planning* (*structures* and) *behaviours* and *control* (*structures* and) *behaviours* of a *secretary* ($U_{PC\text{-}SOA}$) and *office devices* ($D_{PC\text{-}SOA}$) of the *secretarial worksystem*, such that when these behaviours interact with the *perception behaviours* and *execution behaviours* of the *secretary* ($U_{PE\text{-}SOA}$) and *office devices* ($D_{PE\text{-}SOA}$), they *provide support* for *multiple organization communications*, with an *actual level of performance* ($P_{PCMT\text{-}SOA_A}$) which falls within some *desired level of performance* ($P_{PCMT\text{-}SOA_D}$).

HCI-PCMT-SOA design problems may be expressed as:

specify $\{U_{PC\text{-}SOA} \times D_{PC\text{-}SOA}\}$
such that $\{U_{PC\text{-}SOA} \times D_{PC\text{-}SOA}\} \times \{U_{PE\text{-}SOA} \times D_{PE\text{-}SOA}\}$ produces $P_{PCMT\text{-}SOA_{A \to D}}$

where (following on from Axiom 2.1):
-*SOA* denotes secretarial office administration
$P_{PCMT\text{-}SOA} = fn\ \{Q_{MT\text{-}SOA}, K_{PC\text{-}SOA}\}$

Axiom 3.2. HCI-PCMT-SOA domain: organization communications: The secretarial domain has three levels of description: Abstract Level 2, Abstract Level 1, and a Physical Level (see Figures 2 and 4).

At the highest level of description (Abstract Level 2), the *secretarial domain* comprises multiple abstract *Organization Communication* (*OC*) *objects*, which may require temporally overlapping state transformations.

At Abstract Level 1, each OC object comprises an abstract *message object* and an abstract *participants object*. The message and participants objects have *part-of attributes* whose values specify the *OC* object of which they are a part.

At the Physical Level of description, abstract message and participants objects are realized by physical objects. Abstract Level 1 objects have *realization attributes*, whose values specify the physical objects which are their realization.

Physical objects are of two types:

level of description	object	attribute
abstract level 2	organization communication object	*fidelity* *timeliness* *connectedness* *compatibiliy* *comprehension* *participant control*
abstract level 1	message object	*part-of* *body realization* *header realization* *body content* *header content* *access* *channel conformance* *decoder conformance*
	participants object	*part-of* *senders realization* *receiver realization* *intended time scale* *intended receivers* *intended senders* *knowledge of message* *knowledge of participants* *receiver decoding ability*
physical level	agent object	*identity* *location* *[knowledge]* *[intention]* *[skills]*
	simple informational object	*identity* *location* *codes* *format* *quality of encoding*
	composite informational object	*identity* *contained objects* *location*

[ ] Denotes attributes which were not strictly observable, but which are described as physical here as they were not part of the framework's abstraction of entities.

Figure 4. HCI-PCMT-SOA framework: domain objects and attributes.

(i) *informational objects*. The realization of abstract message objects are objects which can carry information and which can be transported to different locations. *Simple informational objects* (e.g., sheets of paper, documents, files, disks, audio-tapes) are distinguished from *composite informational objects* (e.g., document - pile, post-item).

(ii) *agent objects*. The realizations of abstract participants objects are autonomous intelligent objects which have intentions to communicate with each other. For example, individuals or parts of organizations.

The secretarial domain attributes are shown in Figure 4 and defined in Table 3 with example values. Values of Abstract Level 2 attributes are not defined in the table, but vary on abstract scales ranging from 'VERY POOR' to 'VERY GOOD'. Positive values of Abstract Level 2 attributes are associated with high task quality of work transformations.

The realization relationships between Abstract Level 1 objects and Physical Level objects may take the form of one-to-one, one-to-many of many-to-one mappings.

Thus, values of physical object attributes determine, through emergence, the values of Abstract Level 1 attributes (of messages and participants objects). In turn, values of Abstract Level 1 attributes determine, through emergence, the values of Abstract Level 2 attributes (of OC objects). (Vertical relationships are indicated in the attribute definitions of Table 3.)

Composite informational objects, at the Physical Level, have a *contained-objects attribute*, whose value specifies the informational objects, included in the composite object.

Horizontal relationships exist between the values of attributes at the same level of description. For example, at the Physical Level, if a simple informational object is specified, within the value of the *contained-objects attribute* of a composite informational object, then both objects must have identical values of their *location attributes*.

Axiom 3.3. HCI-PCMT-SOA worksystem: The abstract behaviours of the secretarial worksystem are those planning, control, perception and execution behaviours which are particular to effecting secretarial work. The abstract structures of the secretarial worksystem are those planning, controlling, perceiving and executing processes and those knowledge-of-tasks and plan representations particular to supporting secretarial worksystem abstract behaviours.

The physical behaviours and structures of the secretarial worksystem are those which are particular to the embodiment of the abstract behaviours and structures of the secretarial worksystem respectively. The set of physical users and physical devices, which make up a typical *secretarial worksystem* comprises: a secretary (the user) plus office devices such as a word processor, telephone, fax, photocopier, etc.

Axiom 3.4. HCI-PCMT-SOA performance: HCI-PCMT-SOA performance is some function of: (1) task quality associated with the multiple task work of transforming multiple *OC* objects; and (2) the resource costs incurred by the secretarial worksystem behaviours and structures of planning and control.

5. HCI-PCMT-SOA model

This section presents an HCI model of the planning and control of multiple task work in secretarial office administration, based on an

Table 3. HCI-PCMT-SOA framework: object attribute definition and relations. (For each attribute defined, vertical relationships are indicated by listing the related attributes from the next level down. The value of the defined attribute is determined, through emergence, by the combined values of the listed lower level attributes).

abstract level 2 object-attributes

Organization Communication – fidelity

The preservation of message content during the communication.

message-	*body content:*
message-	*access:*

Organization Communication – timeliness

The correspondence between the time of transformation of the message location and the intended timescale.

message-	*access:*
participants:	*intended timescale:*

Organization Communication – connectedness

The correspondence between the intended and actual destination of the message.

message-	*header content:*
message-	*access:*
participants-	*intended receivers:*
participants-	*intended senders:*

Organization Communication – compatibility

The extent to which the message can be decoded by the participants.

message-	*decoder conformance:*
participants-	*receiver decoding ability:*

Organization Communication – comprehension

The extent to which the significance of the message is understood by the participants.

message-	*header content:*
participants-	*knowledge of message:*
participants-	*knowledge of participants:*

Organization Commuunication – participant control

The awareness of the participants concerning the state of the ongoing communication. That is, the correspondence between:

participants-	*knowledge of message:*
and ..	
message-	*all attributes:*
participants-	*knowledge of participants:*
and ..	
participants-	*all attributes:*

abstract level 1 object-attributes

message- *part-of*
The OC object of which the message is a part.

message- *body realization*
The set of physical objects which realize the message body.
Example values: document-1; letter-2
simple IO- *identity*

message- *header realization*
The set of physical objects which realize the message header.
Example values: note-1; envelopt-1.
simple IO- *identity*

message- *body conent*
Semantic information which the message conveys.
Example value: change the pay-status of employee E.
simple IO- *codes*

message- *header content*
Semantic information *about* the message.
Example values: more details will follow; to manager-X.
simple IO- *codes*

participants- *part-of*
The OC object of which the participants is a part.

participants- *senders realization*
The set of physical agent objects which realize the sender of the message.
Example values: manager-P & manager-G; applicants 1..7.
agent- *identity*

participants- *receivers realization*
The set of physical agent objects which realize the receiver of the message.
Example values: manager-D; accounts department.
agent- *identity*

participants- *intended time scale*
The intended time of access changes of the message.
Example values: with P before Friday; with G before the end of today.
[agent- *intention*]

participants- *intended senders*
The set of agents who should send the message.
Example values: manager-G; manager-G and manager-D.
[agent- *intention*]

Table 3 continued over

Table 3. *continued*

message- *access*
The set of agent objects which have access to the message.
Example values: with manager P; with manager G; with manager G (only).
IO- *location*
agent- *identity*
message- *channel conformance*

message- *channel conformance*
The set of communication channels by which the physical realizations of the message may be transmitted
Example values: post (only); email; first-class post & fax; internal post.
simple IO- *codes*
simple IO- *format*
simple IO- *quality of encoding*
composite IO- *contained objects*
composite IO- *location*

message- *decoder conformance*
The set of message decoding systems which are capable of decoding the set of physical realizations of the message.
Example values: english reader & WordPerfect; english reader only
simple IO- *codes*
simple IO- *format*
simple IO- *quality of encoding*

participants- *intended receivers*
The set of agent objects who it is intended should receive the message.
Example value: manager-G & manager X only.
[agent- *intention*]

participants- *knowledge of message*
Values of the message attributes (including projected values) which are known by the participants.
Example values: access: with P (on Tuesday).
[agent- *knowledge*]

participants- *knowlege of participants*
Values of the participants attributes (including projected values) which are known by the participants.
Example values: intended timescale: with P before end of today.
[agent- *knowledge*]

participants- *receiver decoding ability*
Formats of informational objects which the receiver can decode.
Example values: handwriting-english; typing-english; 3.5 disc-WordPerfect -english; printing-english (only).
[agent- *skill*]

physical level object-attributes

simple IO- *identity*
Identity of the object.
Example values: document1; disc2; envelope7.

simple IO- *location*
Physical location.
Example values: out-tray; M-tray; desk.

simple IO- *codes*
Physical codes which encode the message.
Example values: text-characters 1 .. n.

simple IO- *format*
Physical format of the message encoding.
Example values: handwriting.

simple IO- *quality of encoding*
Physical quality of the encoding.
Example values: poor; good.

composite IO- *identity*
Identity of the composite IO object.
Example values: document-pile2; post-item1.

composite IO- *contained objects*
The set of simple IOs and composite IOs which constitute the object.
simple IO- *location*
composite IO- *location*
Example value: envelope1+ address-label1+ letter1+ post-pile1.

composite IO- *location*
Physical location of the composite object.
Example values: out-tray.

agent- *identity*
Identity of the agent object.
Example values: manager-G; manager-M; accounts department.

agent- *location*
Physical location of the agent.
Example values: in M-office; at Heathrow Airport.

[agent- *knowledge*]
[agent- *intention*]
[agent- *skill*]

[ ] Denotes attributes which were not observable, but which are described as physical here as they were not part of the framework's abstraction of entities.

IO-informational object

empirical case-study of secretarial behaviours and work. The model was constructed by interpreting and describing the observed secretarial worksystem in terms of the HCI-PCMT-SOA framework, presented in Section 4. The aim of this section is to show how the framework is instantiated in the case-study, rather than to show how the development of the framework was informed by the case-study. Section 5.1 outlines the methodological details of the observational study of the secretarial worksystem. The resulting model is presented in Sections 5.2, 5.3 and 5.4, which describe the domain, worksystem and performance respectively. The description of the model here is necessarily incomplete, and in some places merely illustrative. [4]

5.1. *Empirical case-study*

The secretarial worksystem selected for study was part of an organization providing recreational services for a large client population. The worksystem comprised a single secretary and the devices shown in Figure 5. It supported the work of managing director *M* of the organization. It also supported related work of other members of the management staff. At the time of study, manager *M* was temporarily holding the office of managing direct *P*. Figure 6 shows the organizational relationship between the agents identified in the case-study.

A 2½ hour video-recording was made of the tasks and behaviours of this secretarial worksystem. On the morning of the recording, managing director *M* was away at a meeting. However, interactions between the secretary and other members of the management staff were observed. During the 2½ hours of recording, the secretary spent roughly half of her time in her own office, either sitting (47%) or standing (7%), and half in another location, either in manager *M*'s office (2.5%), in an outer-office (25%) or elsewhere (18.5%).

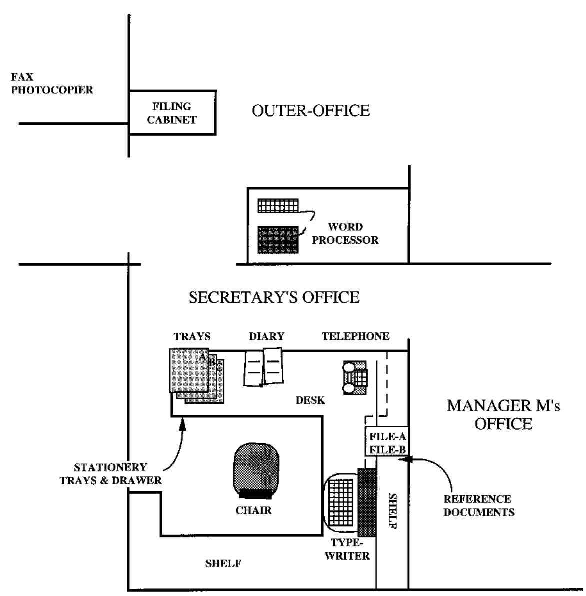

Figure 5. Empirical case-study: office plan.

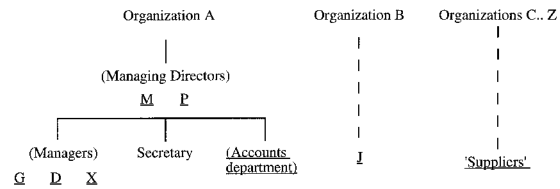

Figure 6. Empirical case study: organizational relationship between the secretary of the observed worksystem and the observed agent objects.

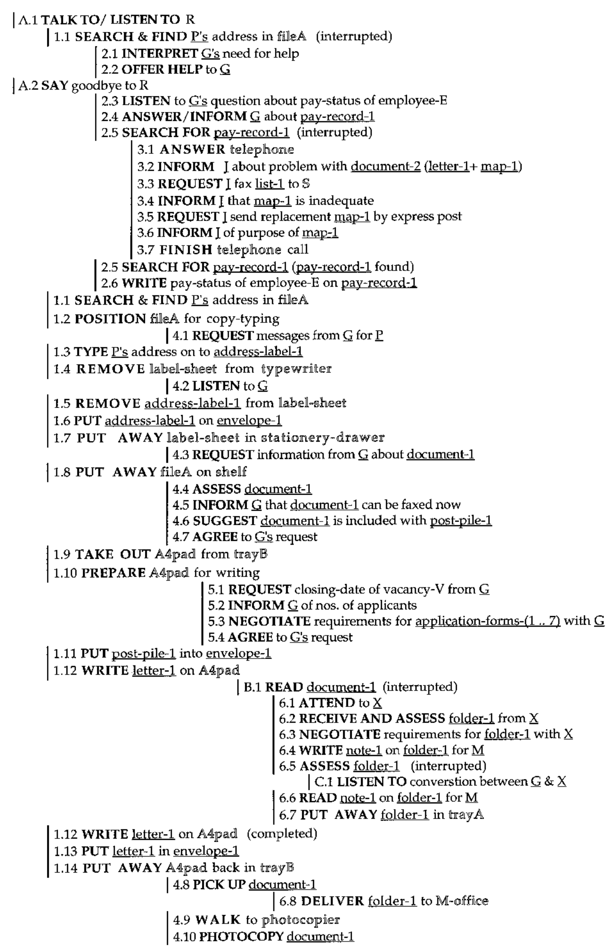

Figure 7. HCI-PCMT-SOA model: interleaved behaviour streams.

A detailed analysis of a continuous 15 minute sequence of video was found to be sufficient for the present purpose of instantiating the framework and informing its development. That is, it was sufficient for the observation of a number of overlapping tasks of the work domain and the associated interleaving of behaviour streams of the interactive worksystem. The particular 15 minute sequence was selected, using the criteria that it was a relatively busy period and that the secretary remained mostly in her own office. Post-recording interviews with the secretary clarified the interpretation of behaviours and tasks.

A 'raw' protocol was first constructed which documented all overt behaviours and their associated physical object transformations. The level of description was thought to be well below that necessary for identifying separate tasks and the associated planning and control behaviours. The secretary was involved in carrying out all of the observed behaviours, mostly in interaction with devices (e.g., *word processor*, *telephone*). Verbalizations of the secretary were recorded verbatim, while non-verbal behaviours took the form of: a physical behaviour (e.g., *MOVE*), specified with reference to associated devices and/or concerned with physical objects, i.e., informational objects (e.g., *document-2*) and/or agent objects (e.g., *manager-G*). A condensed version of the protocol is shown in Figure 7 and described in Section 5.3.1.

The raw protocol was a mixed description of physical behaviours and physical object transformations. It provided the basic source data, concerning both the observed worksystem and its domain of application, to be modelled by the HCI-PCMT-SOA framework.

5.2. *Domain model*

In the raw protocol of physical behaviours and physical object transformations, it was possible to infer six tasks; that is, six Organization Communication objects (OC-1, OC-2, OC-3, OC-4, OC-5, OC-6). The tasks underwent partial transformation effected by the observed secretarial worksystem behaviours.

Figure 8 shows the relationships between Abstract Level 2 objects, Abstract Level 1 objects and Physical Level objects associated with the six tasks. Task 4, for example, concerned the transmission of the physical informational object *document-1* between the physical agent objects *manager-G* and *manager-P*. The physical objects were the realizations of the Abstract Level 1 objects: message-4 and participants-4, respectively.

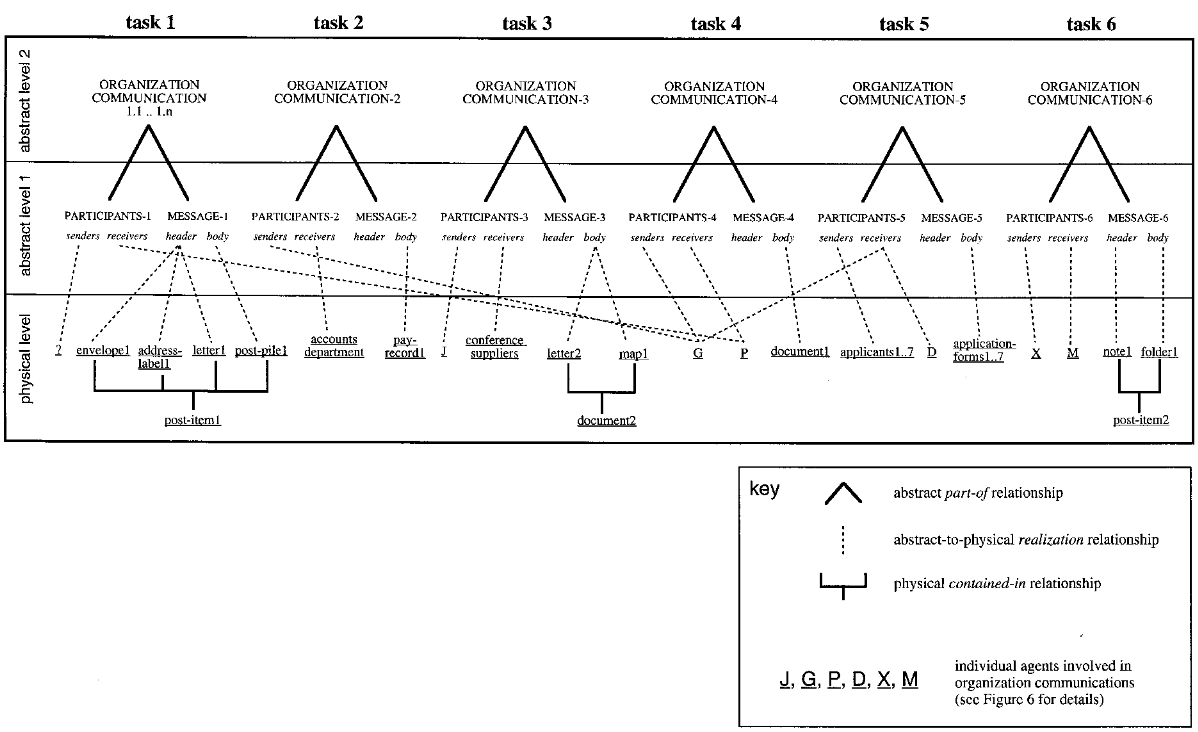

Figure 8. HCI-PCMT-SOA model: domain object relationships.

These Abstract Level 1 objects were part-of the Abstract Level 2 object OC-4. For OC-2, OC-3, OC-4 and OC-5, no Physical Level objects could be identified (within the protocol period) as the value of the message object's *header realization* attribute.

Figure 8 illustrates the different types of mapping between Abstract Level 1 objects and the Physical Level objects:

(i) one-to-one: e.g., participants-6-*senders realization*: to *manager-X*
(ii) one-to-many: e.g., message-3-*body realization*: to *letter-2* and *map-1*
(iii) many-to-one: e.g. participants-2-*senders realization*, participants-4-*senders realization*, and participants-5- *receivers realization*: to *manager-G*

Figure 8 also illustrates relationships between physical objects. For example, in Task 6, the simple informational objects *note-1* and *folder-1* are part-of the composite informational object *post-item-2* . Thus, the value of *post-item-2*'s *contained object* attribute is 'note-1, folder-1'.

Clarification is required concerning OC-1.1 .. 1.n which appears in Figure 8. OC-1.1 refers to the sending of a batch of post, *post-pile-1* , from various senders to *manager-P*. The description OC-1.1 .. 1.n refers to a set of tasks, which have been grouped together in the model. During the analysed sequence, the various senders and the various constituents of *post-pile-1* were not differentiated by the secretarial worksystem. Hence, the representation of the set of tasks OC-1.1 .. 1.n is in the same form as a single task.

Figure 9 shows the attributes of the domain objects transformed by the observed tasks. Attributes were considered to be transformed, if the secretarial worksystem behaviours either changed their values, or maintained them against other influences. Figure 9 also indicates: (1) where the secretarial worksystem carried out (only) perception behaviours relating to particular attributes, and (2) where it could be inferred that the secretarial worksystem had some pre-existing knowledge about an attribute value.

The six observed tasks each concerned only a subset of the total set of domain attributes. However, together they addressed all of the attributes defined in the HCI-PCMT-SOA framework (see Figure 4 and Table 3). Figure 9 also illustrates, for Task 1, the relationship between observed physical behaviours and transformations of Physical Level attributes, and the relationships between those physical attribute transformations and the associated transformations of Abstract Level attributes. For example, the attribute *location* of the physical simple informational object *letter-1* is shown to be transformed by the observed physical behaviour:

1.13 PUT *letter-1* in *envelope-1* (see Figure 7, described in Section 5.3.1).

Transformation of *letter-1*'s *location* attribute (no. 25 in Figure 9) gives rise, through horizontal relation (see Table 3), to a change in the value of the contained *objects* attribute (no. 30) of the composite informational object *post-item-1* . In combination with other physical transformations, which prepare *address-label-1*, this transformation of *post-item-1* gives rise, through vertical relation, to a change at Abstract Level 1. Specifically, a change to *message-1*'s *channel conformance* attribute (no. 13). The *channel conformance* attribute supports, through horizontal relation, a change in the value of the *access* attribute (no. 12) of *message-1*. Finally, the transformation of this *access* attribute gives rise, through vertical relation and in conjunction with other changes, to a transformation at Abstract Level 2 of the *timeliness* and *connectedness* attributes (no. 2 and no. 3,

TASKS	1	2	3	4	5	6
abstract level 2						
Organization Communication objects						
1. fidelity			T			
2. timeliness	T 12,18		T	T	T	
3. connectedness	T 11,12,19			T	T	T
4. compatibiliy		T				
5. comprehension	T 11,21,22					T
6. participant control		T		T	T	
abstract level 1						
Message objects						
7. part-of						
8. body realization			T			
9. header realization	T 24					
10. body content		T	T			
11. header content	T 26					T
12. access	T 13,31		T	T	T	T
13. channel conformance	T 26,27,28,30		T	T		
14. decoder conformance		T				
Participants objects						
15. part-of						
16. senders realization						
17. receiver realization	T			T	T	T
18. intended timescale	K		K	P	P	
19. intended receivers	K			P	P	P
20. intended senders				P		
21. knowledge of message	K	T		T		T
22. knowledge of participt.s	K	T		T	T	
23. receiver decoding ability		K	K			

key
T transformation of attribute value (with the associated attribute changes shown for Task 1)
P perception of attribute value
K attribute value presumed to bc represented in knowledge-of-tasks

physical level (for task 1 only)

Simple Informational objects

24. identity	'address-label-1'	'envelope-1'	'letter-1'
25. location	T (1.5,1.6)	T(1.6,1.11,1.13)	T(1.13)
26. codes	T(1.1,1.2,1.3,1.4,1.7,1.8)		T(1.9,1.10,1.12, 1.14)
27. format	T(1.2,1.3,1.4,1.7)		T(1.9,1.10,1.12, 1.14)
28. quality of encoding	T(1.2,1.3,1.4,1.7)		T(1.9,1.10,1.12, 1.14)

Composite Informational objects

29. identity	'post-item1'	'post-pile1'
30. contained objects	T 25 (1.3,1.6,1.11,1.13)	T(1.11)
31. location		

key T (b1 .. bn) transformation of attribute value and the associated physical behaviours shown in Figure 7

Figure 9. HCI-PCMT-SOA model: attribute transformations and relationships.

respectively) of *organization-communication-1*. This example illustrates the model's description of how physical worksystem behaviours effect transformations of Physical Level object attributes, which give rise, through combinations of horizontal and vertical relations, to high-level abstract attribute transformation. In this example, the physical behaviour of putting a letter in an envelope contributes to the improved timeliness and connectedness of an organization communication.

5.3. *Worksystem model*

5.3.1. *Physical structures and behaviours:* Given the six separate tasks from the raw protocol, the observed physical behaviours of the worksystem were separated into six corresponding behaviour streams. Figure 7 presents a condensed chronological description of the physical behaviours of the worksystem and shows the interleaving of different behaviour streams, resulting from the temporal overlap of the associated multiple task work.

In condensing the sequence of behaviours, all verbalizations were converted to the format used for non-verbal behaviours described in Section 5.1. Almost all of the behaviours fall into numbered behaviour streams 1 – 6 which are task-related and correspond to the numbered tasks shown in Figures 8 and 9. Certain behaviours, however, could not be related to specific tasks and were placed in non-task-related behaviour streams labelled A, B and C. Stream A was a non-work-related social conversation, while Streams B and C were related to work in general, but not to specific tasks. Stream B involved the secretary reading *document-2* to find out its content and Stream C involved the secretary listening to a conversation between *manager-G* and *manager-X*.

The physical structures of the secretarial worksystem were the secretary plus the devices shown in Figure 5: desk, trays; diary; telephone; reference files; typewriter; shelves; stationery; filing cabinet; word processor; fax; and photocopier. The HCI-PCMT-SOA framework supports modelling of the worksystem physical structures only in terms of their identification.

5.3.2. *Abstract structures and behaviours:* The HCI-PCMT-SOA framework supports the modelling of abstract planning and control behaviours for carrying out multiple task work, and their relationship with abstract perception and execution behaviours. Tables 1 and 2 present abstract process structures which were inferred to support the observed planning and control behaviours of the secretarial worksystem. The aim here is not to provide a complete list of such structures which is beyond the scope of a single empirical case-study. Further, it is not claimed that the planning and control behaviours in Tables 1 and 2 were completely implemented in the observed secretarial worksystem. Rather they appeared to underlie the observed sequence of behaviours. The planning and control structures are now described with illustrative examples.

5.3.3. *Control structures:* Control process structures for multiple task work (Table 1) are expressed as *condition-action* production rules. The *conditions* concern the contents of knowledge-of-tasks and plan representations. The *actions* concern the selection of planning, perception and execution behaviours (see Figure 3).

Control Rule 1: plan reading at task finish and natural breaks: When the current task was finished, or when it reached a natural break, the plan representation was consulted to select a behaviour stream to start, continue or resume next. For example, a natural break occurred in Behaviour Stream 5 (see Figure 7) after Behaviour 5.4 (which was the end of the sub-task of establishing *manager-G*'s request). At this point, the previously suspended Behaviour Stream 1 was resumed, prompted by the prominent position of the associated physical objects (*post-pile-1* and *envelope-1*) on the desk. Thus, the arrangement of physical informational objects on the desk is regarded as a type of plan representation, as it can specify information about tasks to be carried out.

The full set of physical plan representations employed by the observed secretarial worksystem (not all exemplified in the analysed sequence shown in Figure 7) were: the arrangement of physical objects on the desk; the positioning of physical objects in trays; self-reminder notes; things-to-do lists; a diary for *manager-M*; and the secretary's memory. Consultations of any of these representations was considered as plan reading.

Control Rule 2: external interruption switching: The external interruptions to which the observed secretarial worksystem responded were visits-in-person and incoming telephone calls. For example, Behaviour Stream 1 was suspended at Behaviour 1.1 for a visit-in-person from *manager-G*. The visit prompted the start of Behaviour Stream 2. Behaviour Stream 2 was then suspended at Behaviour 2.5 for an incoming telephone call prompting the continuation of Behaviour Stream 3 (from an earlier stop prior to the analysed sequence).

Control Rule 3: opportunity switching: The most common type of opportunity was the temporary availability of agent objects, as they passed through, or near to, the secretary's office, or when they contacted the secretary about other matters. For example, Behaviour Stream 1 was suspended at Behaviour 1.10 and Behaviour Stream 5 then continued (from an earlier stop prior to the analysed sequence). This switch was carried out to exploit the opportunity of the temporary availability of *manager-G*, who was questioned about his intention concerning task 5. The application of the opportunity switching rule sometimes follows false alarms in the detection of opportunities relating to the plan. For example, Behaviour Stream 6 was stopped at Behaviour 6.5 to switch to Behaviour C.1 of listening to *manager-G* and *manager-X*. However, in the event, the opportunity failed to progress an existing planned task.

Control Rule 4: suspension avoidance: As implied by Control Rules 2 and 3, the worksystem may switch away from the current behaviour stream in response to external interruptions or opportunities, respectively. On these switches, the worksystem preferred either to finish or to stop the current behaviour stream. That is, to reach a natural break in the current task before switching, rather than immediately suspending the current behaviour stream. For example, the switch from Behaviour 1.4 to Behaviour 4.2 invoked by Control Rule 2, occurred in response to an external interruption from *manager-G*. Although not apparent in the condensed description of Figure 7, the secretary delayed responding to *manager-G*. That is, the secretary delayed switching to Task 4, until the sub-task of typing *manager-P*'s address on to *address-label-1* (involving Behaviours 1.3 and 1.4) had been completed. Thus, the secretary avoided suspending Behaviour Stream 1 in the middle of the sub-task, in preference for stopping at a natural break.

Control Rule 5: plan maintenance: The application of Control Rules 1 and 3 involve the consultation of the existing plan representation. A problem arises if, when consulting the plan representation, it is found to be insufficiently elaborated or scheduled. In this situation, it is necessary to instruct the planning process to develop the plan further, until it reaches a sufficient level of development. Although not explicit in the condensed description of behaviours in Figure 7, an example occurred at the natural break following Behaviour 1.14. Here, the secretary surveyed the desk area and trays and developed her (mental) plans further before continuing Tasks 4 and 6. There were many examples of plan maintenance, elsewhere in the study, which fell outside the analysed sequence. In these examples, the secretary re-arranged documents on the desk and in the trays, wrote self-reminders and constructed plans mentally.

Control Rule 6: knowledge-of-tasks maintenance: On perceiving new information about a task, a sufficient level of detail must be obtained to allow adequate planning. In Behaviours 4.3, 5.3 and 6.3, the secretary actively acquired sufficient levels of detail concerning the requirements for Tasks 4, 5 and 6. Information was requested and clarified concerning, for example, the participants intended receivers of the message and their intended time scales.

Planning structures: Planning process structures for multiple task work (Table 2) are expressed as planning heuristics for specifying the plan representation contents of what or how tasks are to be accomplished. These planning heuristics were inferred as those underlying the formation of plans, which guided the observed behaviours. These heuristics, which are now discussed, can be divided into those which support plan elaboration behaviour (Heuristics 1 and 2) and those which support plan scheduling behaviour (Heuristics 3 and 4).

Planning Heuristic 1: representation of opportunities: For the successful application of Control Rule 3, opportunity switching, it is necessary for the planning process to make explicit potential and relevant opportunities. In the example of opportunity switching, described above under Control Rule 3, the secretary had to make explicit, in the plan for Task 5, the requirement to talk to *manager-G*. Thus, she was prepared to exploit the opportunity of *manager-G*'s availability, when it arose at Behaviour 5.1.

Planning Heuristic 2: sharing behaviour across tasks I – grouping of physical objects: Plan elaboration for multiple tasks sometimes reveals that many similar transformations are required for physical objects relating to different tasks. It may be possible to 'merge' these tasks by first grouping the physical objects into a single composite physical object and then transforming the composite object. Behaviours which transform the composite physical object will progress all of the associated tasks simultaneously. Therefore, these behaviours may be said to be shared across the tasks. The grouping of physical objects must be specified during plan elaboration. An example of sharing behaviours across tasks, through the grouping of physical objects, is seen in the grouping of Task 1.1 .. 1.n (see Figure 8 and Section 5.2). The physical objects, which instantiate the messages of these OCs, have been grouped together into *post-pile-1*. The transformation of *post-pile-1*, therefore, progresses Tasks 1.1 .. 1.n simultaneously. During the observed sequence, the secretary plans to merge Task 4 with Tasks 1.1 .. 1.n. This planned merging occurs in association with Behaviours 4.3 and 4.4, in which the secretary perceives *manager-G*'s intentions and assesses *document-1*.

Planning Heuristic 3: prioritization of suspended tasks: Suspended tasks are prioritized over tasks which have been stopped. This requires the planning process to make explicit their higher priority in the plan. Tasks which have been suspended in Figure 7 (e.g., Task 1 at Behaviour 1.1, and Task 2 at Behaviour 2.5) are all resumed rapidly and in preference to starting or continuing tasks, other than those prompted by external interruptions. In the examples of resumption described in connection with Control Rule 1, it was noted how the worksystem resumed tasks, prompted by a physical plan representation, consisting of the arrangement of associated physical informational objects on the desk. This positioning of documents, in the desk centre, is a form of high priority assignment. In most cases, it is when behaviour streams are suspended that associated informational objects are left prominently on the desk. Thus, their resumption is automatically assigned a high priority.

Planning Heuristic 4: sharing behaviour across tasks II – adjacent scheduling: As noted earlier, plan elaboration for multiple tasks may reveal many sub-tasks, which require similar behaviours or involve similar object transformations. Heuristic 4 advocates adjacent scheduling of these similar sub-tasks. That is, scheduling so that they are carried out in an unbroken sequence. Adjacent scheduling avoids the duplication of set-up behaviours, necessary to prepare for the perception and/or execution behaviours which effect the similar sub-tasks. For example, the observed worksystem carried out Behaviour 6.8 adjacent to Behaviour 4.9. Both of these behaviours required the secretary to leave her office, and their adjacent scheduling required her to leave only once.

5.4. *Performance model*

This section considers the relationship between HCI-PCMT-SOA performance, i.e., the effectiveness with which the secretarial worksystem carried out its multiple tasks, and the planning and control structures for multiple task work. To consider this relationship, it is useful to divide the modelled planning and control structures into three overlapping groups. Each group is a mutually supportive set of control rules and planning heuristics, which in combination implement a particular strategy for achieving better performance. Again, the following planning and control behaviours are not deemed to be an exhaustive set. Rather they represent the modelling of a single case-study.

Group 1: performance and planning

Control Rule 1: plan reading at task finish and natural breaks
Control Rule 5: plan maintenance
Control Rule 6: knowledge-of-tasks maintenance
Planning Heuristic 2: sharing behaviour across Tasks I: grouping of physical objects
Planning Heuristic 4: sharing behaviour across tasks II: adjacent scheduling

This group of planning and control structures concerns the development and exploitation of suitable plans. The observed secretarial domain exhibited some degree of stability over time, which made it effective to plan in advance to some level of detail. By maintaining the plan at a suitable level of detail (Control Rule 5), it could be consulted for guidance at natural breaks in the current task (Control Rule 1). Consultation of the plan ensured that the next task, or sub-task, to be carried out was that which would lead to acceptable performance. Further, the planning process supported the development of plans leading directly to better performance. The two heuristics for sharing behaviour across tasks (Planning Heuristics 2 and 4) produce plans which reduce the behavioural resource costs associated with a group of tasks, or sub-tasks, and/or raise the task quality achieved. To ensure that plans were adequately informed, it was necessary to maintain the knowledge-of-tasks representation at a sufficient level of detail (Control Rule 6).

Group 2: performance and external interruptions

Control Rule 2: external interruption switching
Control Rule 4: suspension avoidance
Planning Heuristic 3: prioritization of suspended tasks

This group of planning and control structures concerns the worksystem's handling of external interruptions. External interruptions reflect some degree of instability over time in the secretarial domain. That is, there were important unexpected events associated with the tasks. To enhance performance, the worksystem switched away from the current task to tasks related to interruptions (Control Rule 2). Failure to switch could seriously reduce the achieved task quality of providing support for organization communications, so reducing performance. External interruptions usually required the worksystem to suspend, rather than stop, the current behaviour stream. That is, to switch from the behaviour stream in the middle of a sub-task, as opposed to switching at a natural break on the boundary between sub-tasks. Suspension is potentially more disruptive for the worksystem's performance than stopping. Individual sub-tasks are usually carried out more effectively in one continuous sequence, thereby avoiding the extra resource costs associated with the repetition of set-up behaviours, including the reconstruction of temporary short-term plans. Thus, a complementary part of dealing with external interruptions was to minimize suspension as far as possible (Control Rule 4), and to maximize the speed of resumption (Planning Heuristic 3).

Group 3: performance and opportunities

Control Rule 3: opportunity switching

Planning Heuristic 1: representation of opportunities

Control Rule 4: suspension avoidance

Planning Heuristic 3: prioritization of suspended tasks

This group of planning and control structures concerns the exploitation of opportunities. To enhance performance, the worksystem switched from the current task to tasks relating to current opportunities (Control Rule 3). To support this switching, it was necessary for the planning process to explicitly represent potential opportunities in the plan (Planning Heuristic 1). As with the response to external interruption, there are some negative effects on performance of switching away from the current task. Reducing the cost of suspension (Control Rule 4 and Planning Heuristic 3) was therefore a complementary part of the overall strategy for handling opportunities. Control behaviours which switch behaviour streams, whether in response to external interruptions or opportunities, must weigh the increased performance on the switched-to task, against the decreased performance on the suspended or stopped task.

The description of the HCI-PCMT-SOA model based on the case-study is now concluded. The purpose has been to illustrate the instantiation of the HCI-PCMT-SOA framework, rather than to present a complete model of planning and control for multiple task work in secretarial office administration.

6. Summary and Conclusions

Design-oriented frameworks are intended to facilitate designers' reasoning about possible solutions to particular HCI design problems. This paper presents such an HCI framework of the planning and control of multiple task work in secretarial office administration. The aim of the paper is to show that the framework is of an appropriate *kind* to support such design reasoning. Validation of the framework will require its application to HCI design practice.

Design-oriented frameworks are for use in a specify-and-implement type of design practice. That is, where solutions to design problems are iteratively specified and implemented, until a desired solution is achieved. The framework is intended to enhance this type of design practice, by reducing the costly specify-implement cycles. This reduction is achieved by introducing relevant HCI knowledge at an early stage of design.

A design-oriented framework aims to provide a designer with the means to create models of specific design problems and to manipulate and compare possible design solutions. Therefore, such a framework must both maintain a general design focus, while supporting the representation of specific instances of design problems and their possible solutions. To maintain design focus, the framework of HCI-PCMT-SOA was developed by the selective expansion of Dowell and Long's (1989) conception of HCI to accommodate descriptions of planning, control, multiple task work and secretarial office administration. To support the representation of specific design problem instances, the framework was also informed by the attempt to model an existing secretarial worksystem and its work domain, and in particular, its performance.

The framework is presented at three levels of generality: HCI, HCI-PCMT and HCI-PCMT-SOA. The development was iterative with successive modifications at the different levels (see Figure 1). Each stage of the development posed the problems of scope and level. What scope should be covered by a design-oriented framework, and to what level of detail should it be described? What scope and level of description, in the HCI-PCMT framework, for example, does a designer require to reason effectively about the specification of the planning and control behaviours for carrying out multiple task work? A conservative strategy was adopted in which concepts were only refined or added to the framework, when necessary, to express the design problem. This approach was intended to preserve coherence by minimizing complexity. Additional iterations of framework-model development made good any deficiencies concerning completeness. Model construction posed similar problems. Here the strategy was liberal, rather than conservative. The result was a model which included, rather than excluded, behaviours where some uncertainty existed. This approach was adopted as an attempt to ensure adequate coverage. Because the HCI-PCMT-SOA model was the lowest level of description (representing an instance), a liberal strategy did not run the risk of over-complexity, which the conservative strategy was intended to obviate.

The application of the design-oriented HCI-PCMT-SOA framework to an observed secretarial worksystem treated the latter as an implemented 'solution' to a design 'problem'. Modifications of the frameworks and model in the light of the empirical observations, during iterations of framework-model development, might therefore be seen as a process of 'backward engineering'. Development worked from the solution to the model, which 'supported' its specification, and then back to the framework, which 'supported' the model construction. The fact that the observed secretarial worksystem was unlikely to have been designed intentionally was not critical. First, the worksystem's planning and control behaviours were presumed to have evolved to carry out work to some acceptable level of performance. Thus, the behaviours might be expected to be of the sort, which could be designed, at least at the high levels of description. Second, the nature of the multiple task work domain is an important part of the model, which exists, independently of the particular worksystem.

The framework is at an early state of development, at least at the higher levels of description. It is offered here as an illustration of an approach to the construction of appropriate design-oriented knowledge. Further development of the framework would seek both to identify and conceptualize new phenomena and to refine the existing concepts. For the domain, development would provide a more complete description of the object attributes, their required transformations and the relationships between those transformations. For the interactive human-computer worksystem, development would provide a more complete description of planning and control structures. For performance, development would provide a more complete description of resource costs, incurred by the worksystem's planning and control behaviours, and the task quality associated with multiple task work.

Although the HCI-PCMT-SOA framework is still in an early form, it nevertheless makes explicit, and therefore supports reasoning about, a specific design solution. Thus, the model in Section 5 shows planning and control behaviours for multiple task work which were observed in an existing secretarial worksystem. The relationship between these behaviours and performance is also made explicit. Using this kind of expression, a designer may reason about possible worksystem configurations with respect to performance, created by the introduction, modification and/or removal of planning and control structures, and their associated performances.

For example, consider Control Rule 1 (see Section 5.3.2, Table 1), which is to read the plan at the finish of each task, and at natural breaks during the tasks. A designer reasoning about this control structure will need to consider the granularity of natural breaks to which the worksystem will respond. A 'coarse grain' worksystem may only consult its plan at the boundary between major sub-tasks. A 'fine grain' worksystem may do so more frequently, at the boundaries between lower level sub-tasks. The most appropriate level of granularity, that is, which produces desired performance, depends chiefly on the stability of the domain. That is, on the frequency of independent changes associated with the tasks. If plan checking is too infrequent relative to domain stability, it will not support the maintenance and use of suitable plans. On the other hand, if plan reading is too frequent, relative to domain stability, the worksystem will incur unnecessary resource costs. This issue of the granularity of natural breaks gives rise to design questions concerning, for example, the specification of computerized planning support tools, as part of integrated office systems. To what extent should the interface allow or facilitate access to the device-based plan during execution and perception behaviours, such as typing letters or reading email messages respectively? Should the interface facilitate plan consultation at appropriate natural breaks in the form of automatic reminders or increased access to the plan representation?

As a second example of how a designer might reason within the concepts of the framework, consider Planning Heuristic 1 (Section 5.3.2, Table 2), which is to make potential opportunities explicit in the plan representation. As in the previous example, the designer needs to specify this structure in greater detail and to reason about its impact on performance. A major design issue concerns the type of potential opportunities to be identified by the worksystem and included in its plan. Failure to be prepared for opportunities will lead to a reduction in task quality. However, effort expended in preparation for opportunities, which never occur, only generates greater resource costs. Again, the optimal preparedness for opportunities will depend on the stability of the domain. In particular, the type and frequency of opportunities which occur. In relation to the computerized planning system described above, examples of design questions here are first, what is a suitable format for a plan which will support the representation of possible opportunities? Second, should the interface promote and facilitate the representation of appropriate types of opportunity, while inhibiting the representation of inappropriate ones? And last, how will the device-based plan support frequent and fast enough reading of opportunities to match them to the user's, or computer's changing knowledge-of-tasks?

A point needs emphasis concerning the generality of the planning and control structures identified in the paper (Tables 1 and 2). While these structures are included as part of the model of the specific secretarial worksystem which was studied (Section 5), they are expressed in a way consistent with the HCI-PCMT framework. That is, they are expressed as independent of both the case-study worksystem and of the secretarial domain in general. This high-level expression of the planning and control structures is deliberate and intended to illustrate how the conceptualization may be more general. Although uncertain at present, it is possible that the current framework captures certain general aspects of the HCI-PCMT class of design problem. These general aspects concern the stability of the domain and its relationship with interruption, opportunities and levels of plan representation. Furthermore, the framework offers a general description of the relationship between planning, control, perception and execution behaviours. This framework embodies an integrated view of planning and control, in which planning behaviours and control behaviours interact to bring about desired levels of performance.

An important challenge in the development of design-oriented frameworks is establishing an appropriate generality of scope. That is, the level of generality of the addressed class of HCI design problem. Low generality tends toward restricted utility. High generality demands greater effort to ensure validity. One strategy is initially to construct a series of frameworks which have low generality, i.e., which address relatively specific classes of design problem. Their combination could then support the construction of more general frameworks. This long-term aim underlies the approach adopted here. Hence, the development and expression of the present framework at successive levels of generality which support increasingly narrow classes of design problem. Preserving the multiple levels facilitates a wider aim of ongoing work to construct a more general HCI-PCMT framework, through related studies of HCI-PCMT-MR (medical reception) and HCI-PCMT-LSP (legal service provision).

Work is also ongoing to develop a 'concept demonstrator' design tool. This tool is intended to show, in principle, how a design-oriented framework might assist designers in generating and evaluating possible worksystem configurations for desired performance. Support would be provided for manipulation of planning and control structures and the prediction of their effects on performance. The initial version of the tool involves a simulation of the HCI-PCMT-MR framework. By demonstrating and investigating how the current framework might support designers, this work aims to establish more firmly the desirable content, format and role of design-oriented frameworks.

Acknowledgements

This work was supported by the UK Joint Councils Initiative (Science and Engineering Research Council; Economic and Social Research Council; and Medical Research Council) in Cognitive Science/HCI, grant no: SPG 8825634. Thanks are also due to three anonymous reviewers.

Notes

[1]See Wallace and Anderson (1993) for an alternative taxonomy of HCI knowledge types.

[2]The more general concept of *devices* here replaces the concept of a *computer* in Dowell and Long's conception. The use of *devices* (which includes computers) allows the framework to model work situations for which computerized support has yet to be developed.

[3]Future versions of the framework may seek to include additional relationships between the worksystem's planning and control structures as shown in Figure 3. For example, the *executing* process might read directly from the *plan*(*s*) representation. The current version of these relationships reflects the empirical case-study reported in Section 5.

[4]A more complete version of the protocol data is available from the first author on request.

References

Ambros-Ingerson, J. A. 1986, Relationships between planning and execution, *Quarterly Newsletter of the Society for the Study of Artificial Intelligence and Simulation of Behaviour*, **57,** 11 – 14.

Bannon, L., Cypher, A., Greenspan, S. and Monty, M. L. 1983, Evaluation of analyses of users' activity organization, *Proceedings of the ACM Human Factors in Software Conference, CHI'83*, 54 – 57.

Beishon, R. J. 1969, An analysis and simulation of an operator's behaviour in controlling continuous baking ovens, in F. Bresson and M. Montmollin (eds). *The Simulation of Human Behaviour* (Duved, Paris), 79 – 90.

Broadbent, D. E. 1971, *Relationship between theory and application in psychology* (Dorset Press, Dorchester).

Card, S. K. and Henderson, A. 1987, A multiple, virtual-workspace interface to support user task switching, *Proceedings of the ACM Human Factors in Computing Systems conference, CHI'87*, 53 – 59.

Carroll, J. M. 1991, *Designing Interaction: Psychology at the Human – Computer Interface*, (Cambridge University Press, Cambridge).

Carroll, J. M. and Campbell, R. L. 1989, Artifacts as psychological theories: the case of human – computer interaction, *Behaviour & Information Technology*, **8,** 247 – 256.

Cherry, E. C. 1953, Some experiments on the recognition of speech with one and with two ears, *Journal of Acoustical Society of America*, **25,** 975 – 979.

Curran, S. and Mitchell, H. 1982, *Office Automation* (Macmillan Press Ltd, London).

Cypher, A. 1986, The structure of users' activities, in D. Norman and S. Draper (eds) *User-Centered System Design*, (Lawrence Erlbaum Associates, Hillsdale NJ), 243 – 264.

Damos, D. 1991, *Multiple Task Performance*, (Taylor & Francis, London).

Davies, P. 1991, Characterizing the program design activity: neither strictly top-down nor globally opportunistic, *Behaviour & Information Technology*, **10,** 171 – 190.

Dowell, J and Long, J. B. 1989, Towards a conception for an engineering discipline of human factors, *Ergonomics*, **32,** 1513 – 1536.

Engel, G. H., Groppuso, J., Lowenstein, R. A. and Traub, W. G. 1979, An office communication system, *IBM System Journal*, **18,** 402 – 431.

GREENBERG, S. 1991, Computer-supported co-operative work and groupware, *International Journal of Man – Machine Studies*, **34,** 133 – 141.

HAYES-ROTH, B. 1985, A blackboard architecture for control, *Artificial Intelligence*, 26, 251 – 321.

HILL, B., LONG, J. B., SMITH, W. and WHITEFIELD, A. D. 1993, Planning for multiple task work: an analysis of a medical reception worksystem, in S. Ashlund, K. Mullet, A. Henderson, E, Hollnagel and T. White (eds) *Human Factors in Computing Systems: Proceedings of INTERCHI'93* (Addison-Wesley, Reading MA), (314 – 321).

HILL, B., LONG, J. B., SMITH, W. and WHITEFIELD, A. D. 1995, A model of medical reception - the planning and control of multiple task work, *Applied Cognitive Psychology*, **9,** 81 – 114.

HIRSCHHEIM, R. A. 1985, *Office Automation: A Social and Organizational Perspective* (John Wiley & Sons, Chichester).

HOLLNAGEL, E., MANCINI, G. and WOODS, D. 1988, *Cognitive Engineering in Complex Dynamic Worlds* (Academic Press, London).

HUGHES, J. A., SOMERVILLE, I., BENTLEY, R and RANDALL, D. 1993, Designing with ethnography: making work visible, *Interacting with Computers*, **5,** 239 – 253.

HUTCHINS, E. 1995, How a cockpit remembers its speeds, *Cognitive Science*, **19,** 265 – 288.

LONG, J. B. 1987, A framework for user models, *Proceedings of the Ergonomics Society Annual conference*, (Taylor & Francis, London).

LONG, J. B. 1991, Theory in human – computer interaction? *Theory in Human Computer Interaction, Institution of Electrical Engineers Colloquium, Digest No: 1991/192*, 2/1 – 2/6.

LONG, J. B. and DOWELL, J. 1989, Conceptions of the discipline of HCI: craft, applied science and engineering, in A. Sutcliffe and L. Macaulay (eds) *People and Computers V: Proceedings of HCI'89* (Cambridge University Press, Cambridge), (9 – 32).

MILLER, G., GALANTER, E and PRIBRAM, K. 1960, *Plans and the Structure of Behaviour* (Holt, Rinehart & Winston, London).

MIYATA, Y. and NORMAN, D. A. 1986, Psychological issues in support of multiple activities, in D. Norman and S. Draper (eds) *User-Centered Systems Design* (Lawrence Erlbaum Associates, Hillsdale, NJ), 266 – 284.

NEWELL, A. SHAW, J. C. and SIMON, H. A. 1958, Elements of theory of human problem solving, *Psychological Review*, **65,** 151 – 166.

NEWELL, A. and SIMON, H. A. 1972, *Human Problem Solving* (Prentice Hall, Englewood Cliffs, NJ).

NEWMAN, W. 1980, Office models and office system design, in N. Naffah (ed), *Integrated Office Systems-Burotics* (North-Holland, Amsterdam).

NORMAN, D. A. 1986, Cognitive engineering, in D. Norman and S. Draper (eds) *User Centered System Design* (Lawrence Erlbaum Associates, Hillsdale, NJ), 31 – 62.

NORMAN, D. A. and DRAPER, S. 1986., *User-Centered Systems Design*, (Lawrence Erlbaum Associates, Hillsdale, NJ).

PAYTON, D. W., ROSENBLATT, K. J. and KEIRSEY, D. M. 1990, Plan guided reaction, *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-20,** 1370 – 1382.

PYLYSHYN, Z. W. 1991, Some remarks on the theory-practice gap, in J. M. Carroll (ed.) *Designing Interaction: Psychology at the Human – Computer Interface* (Cambridge University Press, Cambridge), 39 – 50.

RASMUSSEN, J. and VICENTE, K. 1990, Ecological interfaces; a technological imperative in high tech systems? *International Journal of Human – Computer Interaction*, **2,** 93 – 11.

SCHAFER, G. 1988, *Functional Analyses of Office Requirements: A Multiperspective Approach* (John Wiley & Sons, Chichester).

SCHMIDT, K. 1988, Function analysis instrument, in G. Schafer (ed.) *Functional Analyses of Office Requirements: A Multiperspective Approach* (John Wiley & Sons, Chichester), 261 – 289.

SHIFFRIN, R. M. and SCHNEIDER, W. 1977, Controlled and automatic human information processing II: perceptual learning, automatic attending, and a general theory, *Psychological Review*, **84,** 127 – 190.

SMITH, W., HILL, B., LONG, J. B. and WHITEFIELD, A. D. 1992, Modelling the relationship between planning, control, perception and execution behaviours in interactive worksystems, in D. Diaper, M. Harrison and A. Mong (eds) *People and Computers VII; Proceedings of HCI'92* (Cambridge University Press, Cambridge), 57 – 72.

STORRS, G. 1989, Towards a theory of HCI - a conceptual model of human – computer interaction, *Behaviour & Information Technology*, **8,** 323 – 334.

STORRS, G. 1994, A conceptualisation of multiparty interaction. *Interacting with Computers*, **6,** 173 – 189.

SUCHMAN, L. A. 1987, *Plans and Situated Actions: the Problem of Human – Machine Communication* (Cambridge University Press, Cambridge).

WALLACE, M. D. and ANDERSON, T. J. 1993, Approaches to interface design, *Interacting with Computers*, **5,** 259 – 278.

WHITEFIELD, A. D. 1990, Human – computer interaction models and their roles in the design of interactive systems, in P. Falzon (ed.) *Cognitive Ergonomics: Understanding, Learning and Designing Human – Computer Interaction*, (Academic Press, London), 7 – 25.

WILENSKY, R. 1983, *Planning and Understanding* (Addison-Wesley, Reading, MA).

WOODS, D. D. and HOLLNAGEL, E. 1987, Mapping cognitive demands in complex problem-solving worlds, *International Journal of Man – Machine Studies*, **26,** 257 – 275.

WOODS, D. D. and ROTH, E. M. 1988, Cognitive systems engineering, in M. Helander (ed.) *Handbook of Human – Computer Interaction* (North-Holland, Amsterdam), 3 – 43.

YOUNG, R. and SIMON, T. 1987, Planning in the context of human – computer interaction, in D. Diaper and R. Winder (eds) *People and Computers III: Proceedings of HCI'88*, (Cambridge University Press, Cambridge), 363 – 370.