ANALYSING HUMAN-COMPUTER INTERACTION AS DISTRIBUTED COGNITION: THE RESOURCES MODEL[1]

Peter C. Wright

University of York, UK

Robert, E. Fields

Middlesex University, UK

Michael D. Harrison

University of York, UK

Running head: The resources model

Biographical note:

Peter Wright is a cognitive scientist with an interest in technology, work practice and design methods; he is a lecturer in the HCI group of the Department of Computer Science, University of York, UK.

Bob Fields is a computer scientist with an interest in collaborative work and technology design; he is a senior lecturer in the School of Computing Science, Middlesex University, UK.

[1] As published in Human-Computer interaction Volume 15.1 pp.1-42. 2000

Michael Harrison is a computer scientist with an interest in formal modelling of interactive systems; he is Professor and Head of the HCI group of the Department of Computer Science, University of York, UK.

ABSTRACT

In this paper, we present a new approach to interaction modelling based on the concept of information resources. The approach is inspired by recent distributed cognition (DC) literature but develops a model that applies specifically to human-computer interaction (HCI) modelling. There are of course many approaches to modelling HCI and the motivation of this paper is not to offer yet another approach. Rather our motivation is that the recent developments in DC are so obviously relevant to HCI modelling and design yet the ideas have lacked visibility in the HCI community. By providing a model whose concepts are rooted in DC concepts we hope to achieve this visibility. DC research identifies *resources for action* as central to the interaction between people and technologies, but it stops short of providing a definition of such resources at a level that could be used to analyse interaction. The resources model described in this paper defines a limited number of resource types as abstract information structures that can be used to analyse interaction. We demonstrate how these abstract types can be represented differently in an interface. The resources model uses the concept of interaction strategy to describe the way in which different configurations of resources can differently shape users' actions. These two components of the resources model, information structures and interaction strategies, through the process of co-ordination and integration, provide a link between devices, representations and action that is not well articulated in the DC literature.

CONTENTS

1. INTRODUCTION

Within both the science and engineering of HCI, most models of interaction are task-based. A task is defined as the way in which a goal is attained taking into account factors such as competence, knowledge and constraints. A task representation is a formal abstraction of some aspect of the task such as goals and methods (Card, Moran, & Newell, 1983), task knowledge structures (Johnson, 1992), or the semantic components of a task (Green, Schiele, & Payne, 1988). But the limitations of tasks as a unit of analysis have emerged since HCI research has moved towards the study of computer supported cooperative work (CSCW) (Marti, in press).

Grudin (1990) for example, describes a change in focus in HCI away from the individual user using a single computer, towards groups of individuals communicating through technologies. Rogers and Ellis (1994) argue that traditional task analysis is ineffective for modelling collaborative work and instead argue that a more appropriate unit of analysis is the network of people and technological artefacts involved in the work. In this approach analysis focuses on the transformation of information representations as they are propagated around the network, and also how shared representations are used to co-ordinate collaborative work. This so-called distributed cognition (DC) perspective has been used to describe a range of activities from navigating warships to solving children's puzzles, and DC has been discussed as one component of a theory to bridge research in CSCW and HCI (Nardi, 1996). Yet despite this claim and despite the fact that the DC perspective is so obviously relevant to HCI theory and design, the ideas from distributed cognition research have not really gained visibility in the HCI community.

In this paper, we propose an approach to modelling HCI rooted in DC concepts. We have called our approach the *distributed information resources* model (or *resources* model for short). In many ways the term 'model' is too grand if taken in the spirit of Card Moran and Newell's (1983) work on GOMS, or other formal cognitive models. Rather we mean to use the term to capture the idea that our approach, while incomplete, is systematic and bounded. Perhaps it might be better to think of it as a framework or what Green and Petre (1996) call a "broad brush" approach to interaction analysis.

Our concern is not so much with making model-based predictions of the sort possible with GOMS models but rather with providing a language and a set of concepts for HCI researchers to think about interaction in new ways. With this in mind we do not aim in this paper to validate our 'model' by empirically testing predictions it makes (although it does indeed make some predictions). For us a more appropriate validation will be the extent to which the concepts articulated in the model are picked up and used by HCI designers and evaluators.

The resources model takes seriously the idea introduced to HCI by Suchman (1987), that various kinds of external artefact can serve as *resources for action* and it defines a set of abstract information structures which can be distributed between people and technological artefacts. These information structures can be combined to inform action, and in taking an action new resources are configured. The resources model also introduces the concept of interaction strategy and describes the way in which different interaction strategies exploit different information structures as resources for action. In this sense the resources model is an HCI model in the DC tradition. However, the model is applied here only to examples of single user interaction rather than collaborative work. This is intentional since our aim here is to show how DC ideas can be used in the domain of HCI. The aim of our continuing work is to apply the resources model to collaborative settings where resources are shared between individuals (see Fields, Wright, Marti & Palmonari, 1998 for an example). Before describing the resources model, we review key DC and HCI literature in order to provide a motivation for its development.

1.1. Distributed Cognition

Cognitive science research on the distributed nature of cognition comes from two distinct but not independent intellectual traditions, cognitive anthropology and cognitive psychology. In the former, the emphasis is on the study of real world settings and the role that artefacts play in the practice of work. The research is based on field studies and relates most naturally to the concerns of CSCW. The emphasis of the latter is the study of the individual in a technological environment, and the distinction between knowledge in the head and in the world is central. While everyday work is taken seriously, there is a laboratory tradition at the heart of this approach. Both of these traditions will be reviewed in this section and the commonalities between them drawn out as a motivation for the resources model.

The Anthropology of Hutchins

Hutchins (1995b) is perhaps seen as a chief promoter of DC in cognitive science. As an anthropologist, Hutchins' primary concern has been to consider how cognitive tasks are achieved in everyday settings. He argues that when cognition is studied *in the wild* it is apparent that it is best studied not as an individualistic mental phenomenon, or information processing occurring inside the head of a solitary thinker. Instead it is necessary to consider cognition as a joint activity involving several agents, some human and others technological.

Hutchins (1995a), in a paper entitled *How a cockpit remembers its speeds*, makes explicit the unit of cognitive analysis as a network of people and technologies. The activity studied in the paper is that of an aircraft being flown on its final approach to landing by two pilots supported by a number of technological artefacts. The key problem that the distributed system has to solve is to ensure safe airspeeds during the final approach to landing. As the aircraft approaches the ground it must slow down but as it slows, flaps on the wings have to be extended far enough (but not too far) to maintain the lift required to support the aircraft. There is a minimum safe air speed for a particular flap setting and this varies depending on the weight of the aircraft at the time. Hutchins begins by identifying how information about speeds, weights and flap settings is represented in the cockpit, how these representations are transformed as they flow between agents and also how they are co-ordinated to generate new information and transform the tasks of the pilots. A number of representational artefacts are involved in this task. The mapping between aircraft weights and minimum air speeds for a range of flap settings is represented as a book of look-up tables, which can be positioned above the primary flight instruments, visible to both pilots. This representational device not only transforms a complex arithmetic calculation into a simple look-up task, but the estimate of aircraft weight and minimum speeds on approach to landing is made available for inspection to both pilots for cross checking and later revision if necessary. Once the appropriate airspeeds have been looked up they are further transformed from digital representations to analogue ones by the setting of 'speed bugs' around the dial of the air speed indicators of each pilot. Speed bugs are small slider devices that are moved around the rim of the air speed indicator to mark the position of the safe minimum speeds. Re-representing the digits of the look-up table as positions on the analogue display in this way serves to transform the task of remembering several mode-dependent speeds into a perceptual task of locating the airspeed indicator needle in a particular segment of the dial. This not only provides information about the difference between current speed and minima, but also serves to indicate what configuration the aircraft should be in for the current speed.

In a similar anthropological study on board a US warship, Hutchins (1995b) has shown how tasks such as fixing the location of the ship at sea are achieved by computations distributed across a number of individuals involved in a team navigation activity. As with the aircraft study, much attention is spent on describing how the technology used by the team helps transform analogue to digital representations and how different representations are brought together to generate new information. In addition however, Hutchins explains how the work is organised to maximise error recovery or minimise the impact of error and to distribute different cognitive tasks (e.g. sensing, recording and computing) amongst appropriate individuals. Some detailed consideration is also given to how actions in a shared plan are distributed amongst individuals and synchronised to ensure that the correct information processing steps are taken at the correct time.

As these examples show, a key feature of DC theory in the Hutchins tradition is that the unit of analysis is a network of technologies and actors. Cognition is viewed as a process of co-ordinating distributed internal and external representations to provide a robust information processing system. While Hutchins' emphasis has been on team work, others from cognitive psychology have taken a DC approach to individual problem solving, where the emphasis has been on how cognitive components of a problem are distributed between a single human problem solver and a single problem solving artefact.

Distributed Cognition in Psychology

Norman (1988) popularised the idea that knowledge might be as much *in the world* as it is in the head. Norman's aim here was to point out that the information embedded in technological artefacts was as important to the achievement of a task as the knowledge residing in the mind of the individual who used that artefact. Furthermore, Norman argued that well designed artefacts could reduce the need for users to remember large amounts of information while badly designed artefacts increased the knowledge demands made on the user. The rhetorical aim of the *knowledge in the head, knowledge in the world* distinction was thus to draw the designer's attention to the implications that design decisions had on cognitive processing. The cognitive engineering approach advocated by Norman is premised upon this view of cognition as distributed between user and artefact (Norman, 1986).

In search of theoretical foundations for distributed cognitive systems, Zhang and Norman (1994) have taken the DC approach back into the laboratory to study human problem solving. Here unlike much of Hutchins' work, the emphasis is not on collaborative team working, but rather on the interaction between an individual and a representational artefact and a comparison of behaviours with different information representations. Zhang and Norman studied performance using the 'white rat' of problem solving, the Tower of Hanoi. In the form used by Zhang and Norman, the Tower of Hanoi is an apparatus comprising three pegs. On one peg is stacked three disks of decreasing size. The object of the game is to move the three disks one at a time from the starting peg to a finishing peg via the intermediate peg. Moves are restricted so that a smaller disk cannot be stacked on top of a larger one.

It is possible to generate many different versions of the Tower of Hanoi problem, all of which have the same underlying structure. These structural isomorphs have been used in the past to study issues such as transfer of learning across formally equivalent tasks and the role of context in problem solving (Simon 1975; Luger 1976). The valuable contribution of Zhang and Norman is not the identification of the *sameness* or abstract equivalence of versions of the problem, rather it is the development of a systematic way of describing the *differences* between them.

Zhang and Norman argue that differences between Tower of Hanoi versions can be captured by identifying whether the rules are represented externally in the problem solving apparatus or internally in the problem solver's memory. They also distinguish between rules that are implicitly represented and rules that are explicitly represented. For example, a rule that a disk cannot be placed on a peg occupied by a larger disk can be explicitly written down in the instructions of the problem. Alternatively, in a version in which containers stack inside each other like Russian dolls, the fact that a larger container cannot be stacked inside a smaller one provides a physical constraint that is an implicit representation of the rule. In their analysis Zhang and Norman measure performance on a range of Tower of Hanoi tasks to discover that problems requiring explicit internal representations of the Tower of Hanoi rules make greater demands on problem solvers than problems where the rules are implicit and externalised in the problem apparatus.

More recently, Zhang has attempted to provide a more substantial model for the analysis of distributed problem solving (Zhang, 1997). The model defines external representations as “the knowledge and structure in the environment, as physical symbols, objects or dimensions, and as external rules, constraints, or relations embedded in physical configurations.” (Zhang 1997, p.180). He goes on to argue that information in external representations can be picked up, analysed and processed by perceptual systems alone. These are contrasted with internal representations such as schemata and productions that have to be retrieved from memory. Most interesting cognitive tasks have what is in Zhang’s terms, an internal and an external component (that is to say, they are distributed problems), and as a consequence the problem solving process involves coordinating information from these representations, to produce new information. Deduction is an example of such a process (where for example, an externally represented piece of information serves as the antecedent to some internally represented conditional rule of action). A traditional cognitive account would argue that such deduction is a matter of internalising the external components, but as Zhang points out, this is not always necessary.

> "One view is that external representations are merely inputs and stimuli to the internal mind….Thus, when an agent is faced with a task that requires interactions with the environment, the agent first has to create an internal model of the environment through some encoding processes, then perform mental computations on the contents….A radically different view, offered by Gibson (1966, 1979), is that the environment is highly structured…To Gibson, the information in the environment is sufficient to specify all objects and events in the environment, and thus is sufficient for perception and action. (Zhang, 1997, p.181.)

Hutchins perhaps makes the point more succinctly when he concludes,

> “As we have seen, a good deal of the computation performed by a navigation team is accomplished by processes such as hand-eye co-ordination....The task of navigation requires internal representations of much less of the environment than traditional cognitive science would have led us to expect.” (Hutchins, 1995b, p. 132)

Two key conclusions from Zhang’s work are firstly that appropriate external representations can reduce the difficulty of a task by supporting recognition-based memory or perceptual judgements rather than recall, and secondly, that certain kinds of externalisation can trigger inappropriate problem solving strategies or inferences. This latter finding was further developed in Zhang (1996). This paper identifies a general class of displays including alphanumeric, graphical and tabular displays which he calls Relational Information Displays (RIDs). RIDs are "displays that represent the relations between information dimensions." Zhang categorises information dimensions into four types, nominal, interval, ordinal and ratio, each with characteristic properties forming a hierarchical structure. For example, nominal scales have the property that each item can be categorised as the same or different to any other item, whereas ordinal scales have the nominal property plus the property that each item can be ranked (greater than/less than, next to etc.). Using as an example the desktop display of a Macintosh computer, Zhang compares the dimensional properties of represented information (for example, file size) with the dimensional properties of the representing display. The represented information structure is categorised in terms of the number of dimensions and the scale types of each dimension. This is then used to classify different types of display. So for example, an underlying information structure which has two dimensions, one of which is ratio and one of which is nominal, can be represented by a pie chart or a bar chart.

Zhang then goes on to consider how different information structures support general information processing tasks. For example, three general tasks are: information retrieval (e.g., determining the size of the file called *work*), comparison (for example, assessing whether the file *final* is larger than the file *work*), and integration (for example, integrating information about breadth and depth to determine area). Zhang argues that although there is no task-independent way of determining the best display, there is a general principle that can be derived from the RID framework, namely that "the information perceivable from a RID should exactly match the information required for a task, no less and no more." If the display does not obey this so-called "mapping principle" then the task cannot be achieved unless the user has other information internally represented.

The mapping principle is useful in highlighting that displays are not meaningfully evaluated in a task independent way, and also in highlighting the relation between externally represented information and the residual information required to complete some task. It is usually the case, of course, that displays have to support more than one task, and hence need to be evaluated in the total task context.

Distributed Cognition in HCI

Apart from Zhang's work on RIDs, there have been few explicit attempts to use the ideas of distributed cognition to account for HCI phenomena. Scaife and Rogers (1996) analyse graphical representations as forms of *external cognition* and emphasise the importance of considering how the properties of such representations can affect thinking and reasoning. But they do not provide an account of action or interaction.

In the late 80's there were however, a series of papers published in the HCI community that raised questions about the adequacy of traditional task-based models for analysing interaction with graphical user interfaces. Although not concerned with the ideas of distributed cognition per se, these papers were concerned with how much of the knowledge required for the control of interaction was represented internally by the user and how much was represented as information in the display. Mayes, Draper, McGregor, and Oatley (1988) carried out a series of experimental studies to assess users' knowledge of the menus of popular Macintosh applications. Their studies showed that even expert users could not necessarily recall the names of menu headers yet they had no particular difficulties using the menus in routine work. This led Mayes et al. to conjecture that perhaps users did not routinely commit this type of information to memory but instead relied on cues in the display to trigger the right menu selections. As Young, Howes and Whittington (1990) point out, such a conjecture challenges the assumptions of the well known GOMS family of interaction models (Card, Moran & Newell 1983; Kieras & Polson 1985). The control of interaction in these models is achieved largely through knowledge structures internal to the user. The display does not act as a source of control information. From the Mayes et al. study however it is clear that the display can have a much more central role in controlling interaction in graphical user interfaces. This observation spurred a number of attempts at modelling what has come to be called display-based interaction.

Howes and Payne (1990) developed a model of display-based interaction based on task-action grammars (Payne & Green 1986). They showed that the kind of results found by Mayes et al. (1988) could be accounted for by a grammar that represented semantic but not lexical information about menu headers and other display items. The correct choice of menu items was achieved through a process of semantic matching between the attributes of the task and the semantic features of the display items. Nowhere in the grammar were the actual names of menu headers represented. The model also went some way to simulating typical performance profiles such as scanning menu headers and some menu choice errors.

Using connectionist networks, Kitajima and Polson (1995) simulated the performance of expert users of a graph drawing application. Here too their concern was to model display-based interaction, but in particular they were concerned to account for two related observations. Firstly, that experts often make errors in menu selection and secondly, that they have to deal with large information displays in which much of the information is irrelevant to the task at hand. Display items are represented by arbitrary identifiers and attributes capturing their physical appearance. This information is then used to trigger the user's stored semantic knowledge of display items, including associations between display items and task goals. This so-called elaboration process yields display objects that are relevant to the task at hand and an execution stage selects appropriate actions that are associated with the chosen objects. In a series of simulation experiments, Kitajima and Polson showed that the network made errors and was also capable of recovering from errors. Errors arose for a variety of reasons including the inappropriate selection of objects and recovery could occur because there was no action strongly associated with that incorrect choice. They showed that the error profiles were sensitive to certain parameters of the network. They did not however compare these theoretical results with empirical data.

The work of Howes and Payne (1990) and Kitajima and Polson (1995) is concerned with expert performance. Young, Howes and Whittington (1990) attempted to characterise differences between the display-based competence of novice, intermediate and expert users. Their analysis takes the form of a characterisation of the classes of knowledge required to achieve the task of opening a file on a Macintosh interface. They account for differences in the performance of hypothetical users by appealing to the relative contributions of display information and internalised knowledge in the various classes.

For the expert user, Young et al. follow Howes and Payne in arguing that these users do not possess specialised lexical knowledge of menu names, rather they proceed by semantic matching using translation knowledge and decomposition knowledge that they bring to the task. In contrast, novice users do not possess this expert knowledge and must instead proceed on the basis of lexical characteristics of display items and, through a process of interpretation, make a choice that is the closest approximation to task goals. This choice will be informed by the user's knowledge of everyday meanings of lexical items such as file and edit. The hypothetical intermediate user described by Young et al possesses the translation knowledge of the expert but lacks decomposition knowledge about the ordering of sub-goals. Thus while she can proceed by semantic matching at each step of the task, she will have a greater number of possible matches to consider at each step.

The three approaches to display-based interaction described above represent very different models at different levels of maturity, but they all attempt to characterise display-based interaction in terms of an interaction between information that is external to the user and knowledge that the user brings to the interaction. In this sense, they can be thought of as models that reflect elements of the distributed cognition approach. As Monk (1999) has pointed out however, these models of display-based interaction represent only one kind of interaction. By developing a framework for classifying different interaction models, Monk demonstrates that display based models are unable to model certain types of interaction. The example he cites is that of interaction with moded interfaces. Since control of action is entirely relegated to the display in these models, they would be unable to model error-free interaction with an interface possessing hidden modes. In order to avoid mode errors in these kinds of interfaces the user's internal memory for the effects of previous actions must also play a role in controlling interaction. In order to capture a broader range of interaction styles a more generic approach to the role of display information in supporting interaction is required, one which is capable of spanning a range of interaction styles. The resources model described in the next section aims to achieve this by a more considered use of the ideas from distributed cognition described above.

1.2. Motivation for the Resources Model

Those aspects of DC literature reviewed above provide an indication of the breadth of application of the basic ideas of this research paradigm. Zhang's work has focussed on the individual in laboratory settings while Hutchins' work has focussed on collaborative work.

The display-based interaction work has focused on a very specific aspect of externalised information. Yet all areas of research share commonalities, particularly the emphasis on external representations and the role that these play in transforming work. For modelling HCI, the DC paradigm has some obvious attractions. It might be used to understand how properties of objects on the screen can serve as external representations and reduce cognitive effort (Scaife & Rogers, 1996). It can also serve to bring work on CSCW and HCI closer together by considering how technology mediates the propagation of representations between individuals. The deliberate softening of the boundary between the user and system inherent in the DC view also brings into focus the design question of the information requirements for interaction. What information is required in order to carry out some task and where should it be located? As an interface object, or as something that is mentally represented by the user? As we have seen, there is very little HCI work in this style. The display-based models reviewed above provide only a narrow view on the possible role that external representations can play in supporting interaction.

But as it stands, the DC paradigm has some limitations as an approach to analysing HCI. Central to such a model there needs to be an account of action. We see this in all HCI models. For example, the GOMS concept of operations ties goal representations to action (Card, Moran & Newell, 1983), and in a less formal account, Norman's execution-evaluation loop (Norman, 1988) links the mental representation of plans to actions and their effects on the world. In contrast, Zhang's (1996) RIDs framework is concerned more with inferences drawn from representations, and tasks are seen as something supported more or less well by representations. Indeed Zhang's notion of task is very specific to information search and does not connect with tasks more familiar to HCI. For Zhang and Norman's study of the Tower of Hanoi problem, the link from representation to action is through a notion of planning or plan construction. In their analysis of the Tower of Hanoi problem they identify how rules that constrain the choice of action might be externalised in the device, but the way in which plans (as steps in the solution) could be externalised in the device is not discussed. Hutchins' account of how a cockpit remembers it speed emphasises representations of the state of the world (the speed and configuration of the aircraft and its proximity to landing for example) and goals to be achieved (keeping the airspeed above a certain limit for example). But how these state representations inform the actions of the pilots is less explicit.

Elsewhere, Hutchins has considered representations of plans for action in the form of written procedures for achieving tasks. He has examined how these are distributed in artefacts and how they are internalised by people (Hutchins, 1995b). Within artificial intelligence research, plan construction and plan-following are paradigmatic of the link between representation and action. But in AI as with the GOMS style of modelling a plan is seen as a control structure or program that is internal and usually deterministically followed. The alternative view put forward by Hutchins and more forcefully by Suchman (1987) is that plans *are resources for action* rather than program-like control structures. To make sense of plans as resources for action, plans themselves must be seen as possible objects of conscious attention that can be manipulated, externalised and evaluated.[2] Viewed in this way, plans can, like RIDs, be seen as distributed representations. But where RIDs are representations of state information, plans are representations of possible courses of action. Similarly in display-based interaction, pull-down menus can be seen as resources for action in the process of goal-matching.

In pursuit of a DC model of HCI based on the concept of resources for action, we have identified and defined a number of resources for action and explored how different system implementations distribute these resources between people and technologies. Like Zhang and Norman's Tower of Hanoi rules, we have also observed how these resources are sometimes explicitly represented and sometimes less so. In this model, which we refer to as the resources model, Zhang's RIDs and Suchman's plans are but two of a number of possible resources for action. One of the consequences of the resources model is that different styles of interaction, or what Zhang might refer to as problem solving strategies, emerge as a consequence of how the interaction is resourced in terms of what resources are available and how and where they are represented for use.

---

[2] It is not our intention here to suggest that all control of action must necessarily be deliberate, or available to introspection. This is clearly not the case. Rather we mean to suggest that plans, intentions, etc. can be reasoned about, altered, communicated to others, diverged from, etc. Neither do we mean to suggest that even as conscious objects they are necessarily formulated prior to engaging in action. For example, Suchman argues that plans may emerge from interaction in a situated way (see Suchman, 1993).

2. THE DISTRIBUTED INFORMATION RESOURCES MODEL

The resources model (Wright, Fields & Harrison, 1996; Fields, Wright & Harrison, 1997) has two central components. The first is a characterisation of information structures relevant to the control of action. The second is a more process-oriented characterisation of how these information structures can be used as resources for action.

2.1. Information Structures

The resources model distinguishes between abstract information structures on the one hand and their representation (or implementation) in an interaction on the other. This distinction follows that of Marr (1982) and Hutchins (1995b). The abstract level of analysis allows us to consider the structural characteristics of the information that are essential for them to serve as resources of various kinds. It also allows us to consider certain kinds of equivalence between different representations of information. The representational level allows us to consider the detail of how an information structure is distributed between people and artefacts and the form and content of the external aspects of the representation.

Abstract Information Structures

The resources model identifies six information structures that can be defined at an abstract level in the same way that Zhang (1996) defines the RIDs taxonomy and Zhang and Norman (1994) define the information structures of the Tower of Hanoi. That is to say, the type and structure of the information can be described independently of how it might be represented in an information artefact. In this section we will first describe the abstract information structures which constitute the building blocks of the resources model. Secondly, we will give examples of the different ways these abstract information structures can be represented for use as resources. The abstract information structures are as follows:

- Plans
- Goals
- Possibilities
- History

- Action-effect relations

- States

These types have been derived from the HCI and CSCW literature in an inductive way. We have found these six basic structures sufficient for the kinds of analyses we have so far carried out and believe they form a coherent core for HCI modelling. However, it is entirely possible that more structures could be required.

Plans

The idea of a plan as a resource for action comes from the work of Suchman (1987) and Schmidt (1997). In the resources model, a plan is a sequence of actions, events or states that could be carried out. Plans can involve conditional branches and loops that are dependent on the *state* of the system (see below). When using a plan it is important to know where one is in the plan. This can be achieved for example by combining the plan with a representation of the history of actions so far carried out (see below).

Goals

Goals, like plans, are traditionally viewed as solely mental constructs, but within the resources model goals, like plans, can be viewed as an abstract information structure. At this abstract level a goal is a required state of the world. Monk (1999) has argued the value of associating goals with states of the system in this way.

Possibilities

In the resources model we use the term *possibilities* to refer to the *set* of possible next actions that can be taken by a user for a given state of the system.[3] This set has to be defined with respect to a particular level of analysis. In this paper, a set of possibilities at a

---

[3] In previous papers we have used the term *affordances* rather than possibilities. This has proved confusing to some readers and so we have changed our terminology. The term affordance was introduced to psychology by Gibson (1966, 1979) and has since been used by others in the HCI literature. For many, the term has come to refer to some intuitive way of using an artefact. So for example, certain kinds of door handle are intuitive in that they afford pushing not pulling or vice versa. Recently Norman (1999) has argued with this view re-asserting instead Gibson's original intended meaning as the "Actionable properties between the world and an actor" (Norman, 1999). This is definition is closer to our intended meaning for possibilities.

functional or logical level might be the set of commands that a computer is capable of responding to in a given state.

History

An interaction history, like a plan, is a *sequence* of actions, events or states. Unlike a plan however, the sequence comprises those actions, events or states already achieved in the interaction and therefore it does not contain any branching or looping. As Monk (1999) points out, history plays an important role in some interactions. The example he offers is as a means of determining modes in systems where mode changes are not signalled in the display.

Action-effect relations

The idea of action-effect relations comes from work of Monk (1990). An action effect relation is a causal relation between an action or event and a state which represents the effect that executing the action or event will have on the interaction. The set of action-effect relations for a given state is referred to as an action-effect model for that state.

States

The state is the collection of relevant values of the objects that feature in the interaction at a given point in the interaction. Like possibilities described above, the set of states has to be defined with respect to a particular level of analysis.

Representing Abstract Information Structures

Before information can be used as a resource for action it has to be represented during the interaction. A given information structure may be represented externally in the interface, internally in the head of the user, or more often, distributed across the two.

Plans

Plans can be represented internally as memorised procedures to complete some task. They can also be represented externally as standard operating procedures, or step-by-step instructions for achieving some goal. If a person is following a plan, it is important to know exactly where in the plan they are. The sequence of actions constituting the plan can be externalised in written a procedure, in which case, one's current position might be represented by a thumb or finger placed alongside the last action or perhaps by ticking off those actions already done (See Hutchins, 1995b and Figure 4a for examples).

Goals

Goals can also be distributed across artefacts and people. For example, aircraft flight control interfaces often include something called a *flight director*. This indicates to the pilot which way she must steer in order to continue on the desired flight path. It is not a plan because it does not describe the actions required to get the aircraft on the right path. It is a goal because it specifies a state of the world to be achieved.[4]

Possibilities

Two everyday examples of externally represented possibilities are menus (both on computers and in restaurants) which represent the set of things that can be chosen, and road junctions which represent the set of possible routes to take. In both cases the artefact or situation represents the set of possible actions. For some interactions, possibilities may not be externally represented. Consider for example a command line interface to UNIX used by UNIX experts. There is nothing externalised in the display that indicates the possible commands that can be issued. They are represented only in the head of the expert user.

History

As an example of history, consider again the UNIX command line interface. The interface displays some of the previous commands that the user has issued thus externalising some aspects of the history of the interaction. As another example, *Undo* commands in word processors and *back*, *forward* and *go* commands in web browsers, all rely on some representation of the interaction history. But the history is not always displayed to the user.

Action-effect relations

Internal representations of action-effects can be generated by a user's conceptual model of a system, but they may be externalised in user manuals or as part of the help system for the

---

[4] More precisely, the relation of the represented position of the aircraft to the represented target position, in the context of an intention to follow a route, generates for the user of the resources, a goal to close the gap.

user interface. The prototypical form of this information is a condition-action pair that says "if you do this then the following will happen".

State

As an example of a state that is externalised, consider the Tower of Hanoi problem described earlier. The current state is represented as the positions of the disks on the pegs. Similarly, in Hutchins' cockpit example described earlier, the values of the flap and slat settings and the current indicated air speed are relevant aspects of the current state externalised for the user.

2.2. Processing in the Resources Model

In the previous section we have described some of the types of information structure that can serve as resources for action. In this section we take a process view and describe how the resources represented in an interaction are used to inform action. We first describe a cyclic model of interaction appropriate for our needs and then discuss how different configurations of resources can be used in conjunction with different interaction strategies to generate action.

Cyclic Models of Interaction

As Monk (1999) points out, it is useful to think of human-computer interaction as a continuous process of cyclic interaction with the environment involving *recognition* of changes in the state of the environment and action on the *environment* to bring about changes. In the classic cyclic model of Norman (1986) for example, changes in the environment are *perceived*, *interpreted* and *evaluated* with respect to *goals* and this allows for *plans* of action to be generated and *executed* leading to changes in the state of the world.

The display-based models described above (e.g. Howes & Payne 1990) and the GOMS family of models are also examples of cyclic models (Card, Moran & Newell 1983). Monk argues that while such models are predicated on a clear division between internal and external (user and environment), few provide explicit models of the environment. Monk also distinguishes between models that emphasise control flow, and those that emphasise information flow. As an example of the former, he cites the GOMS model again. The model requires a goal to be specified and the model characterises the control structures required to decompose the goal into sub-goals as a set of production rules. In display-based models the decomposition into simple actions is taken for granted and the emphasis is placed on the role that aspects of external cues play in cycles of recognition, decision and action.

The resources model is also a cyclic model but the distinction between user and environment and between information and control flow is de-emphasised. Plans as control structures can for example be external or internal to the user or distributed between user and system. The model is cyclic in the sense that action is informed by the configuration of resources represented in the interaction at any particular time, either externally in the interface or internally in the head of the user. When an action is taken the *configuration* of resources is changed. Minimally for example, the history of the interaction always changes when an action is taken. This new configuration of resources is then used to consider the next action. This cyclic interaction is illustrated in Figure 1.

<Figure 1 about here>

By configuration of resources we mean the collection of information structures that find expression as representations internally or externally at a given step in the interaction. In using the term *action* we do not confine ourselves to single actions such as clicking on an icon. At a given point in the interaction the resource configuration may be sufficient to determine a whole sequence of such primitive actions. An interaction sequence can thus be described as a number of steps from one *resource configuration* to another. We also make a distinction between our primary concern which is a *logical* action such as choosing a menu item and the physical instantiation of that action such as moving a mouse cursor to a particular area of the screen.

The decision as to what logical action to take involves bringing together these different representations through processes which Hutchins describes as *co-ordination*. Co-ordination processes are discussed in great detail by Hutchins (1995b), but in this paper we shall use the term in a relatively restricted sense. How easy it is to co-ordinate one or more resources is determined in part by the location and physical attributes of their implementation. For example, a restaurant diner may require both the menu and a memory for the meals she has chosen previously in order to decide what to eat. The memory for previous choices is a representation of history and the menu is a representation of possibilities. The history is an internal resource and the possibilities an external one. The process of co-ordination in this case requires our diner to read and understand the menu, to be able to recall accurately enough the names of previously eaten meals and to determine whether items in the menu have been tried before.

Interaction Strategies

The second part of the processing model is concerned with how resources can be used to inform action. A configuration of resources can be used in different ways. People interact with the same graphical user interface in a variety of ways such as the display-based interaction referred to in our review above. We use the term *interaction strategy* to describe different ways in which resources can be used to make decisions about action.

Within the cognitive psychology literature there have been many demonstrations of the way in which people adapt their problem solving strategies to exploit constraints and opportunities in the problem solving environment (Sternberg & Wagner, 1994). To take an everyday example, a learning test which adopts a multi-choice answer format allows the student to answer questions by directly recalling the answer from memory or by eliminating the implausible responses from the set of responses offered. The success of the elimination strategy will depend on the details of the responses offered. Thus the ability of the student to answer the question is a product not only of his learning but also of the details of the test design. In the case of HCI a similar distinction can be made between command line interfaces which require the user to recall appropriate commands and menu-based interfaces which support a range of interaction strategies including eliminating implausible options.

Interaction strategies presuppose certain configurations of resources to make them effective and conversely a particular configuration of resources makes particular interaction strategies possible. Below we describe a range of interaction strategies relevant to HCI and relate these to the information resources they require. Like our abstract information structures, we do not suppose that this is an exhaustive list. Rather it is sufficient for our current purposes.

Plan following

As an interaction strategy, plan following involves the user in co-ordinating a pre-computed plan with the history of action so far undertaken. In its simplest form the plan is followed by determining the next action on the list until the list is exhausted. Some plans may have conditional steps requiring the plan follower to examine the current state of the system. Plans are often followed for a particular goal but it is possible to follow a plan blindly without any knowledge of what it will accomplish.

A pre-computed plan is central to the plan following strategy. Thus a plan-following interaction will have the plan as a resource either externalised in the interface (see Figure 4a & 4b), recalled by the user or recorded in some other form (as an operating procedure kept in a manual for example). In the plan following interaction, the plan and interaction history need to be maintained and co-ordinated in order to keep a sense of position within the plan. State and goal resources may also need to be co-ordinated with this position to deal with conditionals and to assess whether the goal has been achieved.

Plan construction

Plan following requires the use of a pre-computed plan. A more elaborate activity may involve an initial phase of *plan construction,* followed by a phase of following the plan that has been constructed. While the output of a plan-following strategy is a sequence of actions that is taken, the output of a plan construction activity is the plan for use as a resource. Plan construction typically requires the user to compare the current state of the world with some goal state and to select from possible next actions those that reduce the difference between the two states. This process may need to be done iteratively on future states and possibilities. Plan construction thus co-ordinates four types of information resource: goal, states, possibilities and action-effects.

Plan construction may be a complicated process because as the state changes, the actions that are available or appropriate will also change. This makes plan construction a costly activity if it is done entirely in the head of the user. Resources necessary for plan construction can be externalised however. A system that supports plan construction might for example make the developing plan available for manipulation by the user. It might also and clarify the nature of the changing availability of the actions and their effects. Plan construction itself will require the planner to adopt some form of strategy to construct the plan. Artificial intelligence research provides many such plan construction strategies (see for example Russell & Norvig, 1995).

Goal matching

As our earlier review of display-based interaction showed, in many situations humans do not act by following previously constructed plans, nor by constructing plans to follow 'on the fly'. Instead they use possibilities visible in the display to control a search for a next action that matches a goal. The resources co-ordinated to support this strategy are a goal, a set of possibilities and action-effect relations. Graphical user interfaces provide support for goal matching by providing, in the interface, representations of the commands that are available and the effects that they have.

History-based selection and elimination

HCI has given little attention to the role of history in decisions about action, but as we saw with our earlier story about the restaurant diner, one strategy for choosing among possibilities is to eliminate those that have already been chosen. Alternatively, history could be used to repeat an action that had previously been taken. Interfaces that support these strategies might have some inspectable representation of history such as the *go* function available in many web-browsers. A key feature of web browsers is that an item in the history is also a possibility. That is to say, clicking on an item in the history list takes you there.[5] History-based choice bears a strong resemblance to *trial and error learning* studied by psychologists where one of a set of possibilities is chosen at random and repeated if it leads to a desirable outcome.

The interaction strategy a user adopts will, in part, be shaped by the resources that are available to her. Figure 2 summarises the description of interaction strategies given above and relates these to the abstract resources that are required for their use.

---

[5] Note that in many commonly used browsers, the options available in a *Go* or *Back* menu are not an accurate representation of the interaction history (see Wright, Fields and Merriam, 1997).

<Figure 2 about here>

3. USING THE RESOURCES MODEL TO ANALYSE INTERACTION

In this section we aim to describe three ways that the resources model has proved useful in framing an analysis of interaction in terms of distributed cognition. The first is as a means of comparing different interface designs. The second is as a means of analysing interaction scenarios. The third is as a way of generating design alternatives and analysing their effects on user performance.

3.1. Comparing Interfaces

The way in which resources are distributed between people and artefacts can affect the ease with which an interactional goal can be achieved. This may be reflected in the difficulty of using a particular interaction strategy. Scaife and Rogers (1996) have used the term “computational off-loading” in a general way to describe how certain kinds of external information can reduce the mental effort involved in achieving a task. Similarly Zhang (1996) has argued that tasks involving perceptual judgements can be less demanding than those involving mental arithmetic. Within the resources model these differences are reflected in terms of the difficulty of coordinating resources. In this section we present three examples of how the resources model can be used to characterise the differences between alternative interface designs.

Achieving Target Speeds by Goal Matching

In Hutchins’ (1995a) example of cockpit speeds described earlier, one of the pilots’ tasks is to ensure that the flap settings are appropriate for the current speed and to extend the flaps at an appropriate time before the minimum safe limit is reached. Part of this task involves knowing the 'target speed' at which the flaps are to be extended. It is thus necessary to notice when the declining speed approaches this target, at which point the flap setting should be adjusted.

One of the tasks that is carried out is therefore to make a comparison between a target or goal state (the target speed) and the current state (i.e., the current speed). In order to do this, the goal and current state resources must be brought into co-ordination, and precisely how this happens is highly dependent on the way the resources are represented in the interaction. Figure 3 shows three examples of how the goal and current state may be represented to support this activity; in each case the process of co-ordinating resources to make the task-relevant comparison is different.

<Figure 3 about here>

1. As in Hutchins' real example, the current and target speeds are represented as a pointer and *speed bug* respectively. The process of comparison and co-ordination then becomes a perceptual judgement about the relative locations of pointer and speed bug.

2. The current speed is represented in a numeric display line, and the target is represented internally (i.e., remembered) by the pilot. In this case the process of co-ordination involves the ability of the pilot to read the display, interpret it (employing knowledge about decimal numerals), and make the comparison with the remembered target speed to determine whether the remembered speed is higher or lower.[6]

3. Both current and target speed values are represented in numerical form in the flight deck display, allowing the pilot to make an explicit co-ordination similar to case (b), with the difference that both values are read and interpreted, and neither of them need be represented only in the pilot's memory. In addition, the flight deck computer system also makes a comparison and displays the result as a piece of text in the display.

Following a Plan: Co-ordinating Plans and Histories.

As a second example, consider a task common in safety-critical settings, namely the task of following a procedure (see Degani & Wiener, 1990 for a discussion). Following a

[6] Note that interpreting the numeric display relies on a further co-ordination of information structures (both internal and external) that we will not explore here, but see (Zhang, 1997) or (Hutchins, 1995b) for a detailed discussion.

procedure is an example of the plan-following strategy described above. Plan following is possible when what Hutchins (1995b) refers to as a pre-computed plan exists and the user follows it. In abstract terms the process for selecting the next action to perform is simple: compare the complete plan and the history of which actions have already been performed in order to determine which is the next item on the plan. Once again, the actual work involved in accomplishing this co-ordination is highly dependent of the representational form of the resources. In Figure 4, three illustrations are given of how the nature of plan and history representations determine the kind of co-ordination work that is needed for plan-following.

<Figure 4 about here>

a) The *plan* resource is represented as a printed procedure of actions to perform. The *history* is differentiated by the user of the procedure marking the next item to perform with her finger. The co-ordination simply involves looking for the item next to the finger. Note that this means the representation of the plan, the next action to perform, and the history of actions that have been performed are very tightly integrated.

b) Here the plan, the history and the marker for the next action are all maintained electronically by the computer system and are represented in a single display. Actions performed already (the intersection of the history and the plan) are shown on a grey background, and the next action is shown in reverse type. Again, the co-ordination is trivial: it has already been performed by the machine.

c) This is a variation on (b): both the plan and history are stored inside the machine, but only the next action is actually displayed.

As both Figure 3 and 4 illustrate, in some situations co-ordination can be particularly well supported if the resources that need to be co-ordinated are represented in the same external space. In Figure 3a for example, state and goal are co-located in the airspeed indicator. Similarly in Figure 4a, by introducing her finger as a marker the pilot produces a co-ordination of history and plan in a single external representation.

In Figure 3 we described a number of different implementations of speed information in a cockpit and in Figure 4 we compared a number of different ways to implement procedure following. These examples are indicative in general terms of the kinds of display that can be found in modern commercial aircraft. They are useful in showing how interface implementations can be compared in terms of the different resources they implement. A more detailed comparison of the user interfaces of commercial aircraft systems has been carried out using the resources model (Hicks, Wright & Pocock, 1999). The results of this analysis highlight differences in the way in which history and plan markers are implemented in the interface and the possible implications they might have for human error. Figures 5 and 6 illustrate examples of interfaces more familiar in HCI research.

<Figure 5 about here>

Figure 5 shows an extract of a dialogue from a Microsoft spreadsheet application and Figure 6 a dialogue box from a similar ClarisWorks spreadsheet. In order to complete a chart in the Microsoft application, the user goes through a prescribed sequence of dialogue boxes known as a Chart Wizard by clicking the *next* button after various choices have been made in the box itself. In the ClarisWorks application the same effect is achieved in one dialogue box and the order in which changes and parameter selections can be made is unconstrained. A resource-based account of these differences would point to the support that is given to plan-following in the Microsoft application. The user progresses through a sequence of five dialogue boxes, and these represent a high-level plan for generating the chart. This sequence of dialogue boxes is an externalisation of a plan resource. The implementation does not, however, make the plan explicit to the user since at any one time only a small fragment of it is visible in the form of the current box (cf. Figure 4c for an analogous distinction). This plan is externalised in the application program and it prescribes the order in which actions are taken. The user need have no memory of this plan in order to interact with the application. In ClarisWorks, there is no plan-following resource externalised for the user. The order in which the actions are carried out is unconstrained. For plan following to occur with this interface the plan must be remembered by the user.

An alternative strategy that can be used however is goal matching, based on the meanings of the five ‘modify’ buttons, or an elimination strategy based on previous history.

<Figure 6 about here>

In all the examples given above the differences between the interfaces are easily apparent, yet surprisingly few models of interaction provide the necessary concepts to capture these differences. A GOMS style model would probably show how the goals represented in the ClarisWorks display were decomposed, but it would have little to say about the way in which the Chart Wizard serves to externalise the control of action, requiring no internalisation of a goal decomposition on the part of the user. The display-based models would provide an account of how the ClarisWorks interaction might proceed but it would fail to capture the plan structure that is inherent in the externalised plan of the Microsoft Wizard.

3.2. Analysing Interaction Episodes

As well as comparing across interfaces, the resources model can be used to identify parts of an interaction scenario where the distribution of resources might place heavy demands on coordination and consequently might shape the choice of interaction strategy. As an example, we have conducted a walkthrough of a scenario in which a user attempts to use the Chart Wizard described in the previous section. This example highlights parts of the interaction where errors or breakdowns might occur. It also illustrates the way in which even within a relatively small episode of interaction we might find changes and blends of interaction strategy.

Overview of the Chart Wizard Dialogue

The task of making a chart with the Chart Wizard involves the following:

4. The user must select some data to be made into a chart.

5. The user must then select the Chart Wizard icon.

6. The user must then drag in the spreadsheet to indicate where the chart should be placed.

7. The user must then follow five steps in a dialogue. These involve:

   – selecting or confirming the data to be charted,

   – choosing style of chart (e.g., line or column),

   – choosing a particular version of that style (e.g. 2D or 3D column),

   – specifying the mapping between data and axes of the chart,

   – selecting labels.

For this scenario we assume a user who is a regular user of Macintosh style interfaces and who has some experience with earlier versions of the spreadsheet package but has not previously used Chart Wizard. The user has some data she wishes to chart and this is already selected in the spreadsheet. She is aware that it is possible to produce charts directly from data but does not know how this is done. What we are going to do now is to take a step-by-step walk through the above task, looking at information resources that are present, and the way they are used to guide the course of action that unfolds.

A Step-by-Step Walkthrough

Step 1

At the start of the interaction a number of action possibilities are available to the user, most particularly the Chart Wizard button in the toolbar. In common with many modern user interfaces when the pointer is moved over an icon a pop-up text box appears showing the name or a brief description of the function activated by the icon (see Figure 7). This can serve as an external representation of an action-effect mapping, providing a resource that aids matching between chart-making goals and the possibilities represented at the interface. Figure 7 illustrates the screen at the start of this step.

<Figure 7 about here>

Step 2

After clicking on the Chart Wizard button to start the chart production process, new interaction resources are configured. The spreadsheet window changes so as to appear as in Figure 8. The mouse pointer has changed shape to a cross hair and the border around the selected data has also changed. These external resources, on their own, are insufficient for the user to decide what action to take next. But at the bottom of the screen, an action-effect mapping has been expressed "drag in document to create a chart". This action-effect mapping in the context of the user's goal is the key resource for action and supports a goal matching strategy, however it is easily overlooked placed as it is in the bottom left of the screen away from the users natural focus of attention.

<Figure 8 about here>

Steps 1 and 2 are not supported by an externalised plan. Furthermore, a relatively inexperienced user would not have an internal plan for this task. Consequently, success at step 2 depends entirely on the user seeing and understanding the action-effect information at the bottom of the screen. The resource model would predict that a goal matching strategy would be required here and that novice users would have particular difficulty with this part of the interaction episode.

Step 3

Carrying out the dragging action leads to the screen configuration shown in Figure 9. A dialogue box has now appeared. The appearance of this box marks a significant change in the available resources. The dialogue box is a resource which can be used to support plan following. The number of steps in the plan and the user's current position in it are clearly indicated in the dialogue box. This step triggers a new and unexpected sub-goal for the user, that of selecting appropriate data for charting. In fact the user has already done this before the start of step 1. In order to avoid an unnecessary step of re-selecting the data, the user must have access to the interaction history. This is not represented externally in the interface, the user must rely on her internalisation of history to avoid this unnecessary step. State information expressed on the screen as a box around the data and a range of data co-ordinates indicates which data are currently selected. Nevertheless the user must still remember whether the information so selected is the information she wishes to chart. Note also that once the next step is taken, the box around the data disappears and the information about which data are going to be charted is no longer visible. The user must rely entirely on memory.

<Figure 9 about here>

Step 4

The next step involves defining the nature of the chart to be produced based on the selected data. A number of alternative images are offered suggesting the style of chart that is to be produced (see Figure 10). Highlighting is supposed to create an association with the idea that this is the type of chart to be selected in the context of the rest of the plan. Buttons are offered in order to continue the plan. Other buttons are also offered which indicate their effects of for example, aborting the plan and returning to the beginning or finishing the plan taking all the remaining defaults.

<Figure 10 about here>

Interestingly, the images serve not only as a goal resource indicating what charts are possible but also as action possibilities since clicking on the images will move the user towards the chosen goal. As well as being expressed implicitly as a gallery of images, the new sub-goal is also expressed explicitly by the prompt ‘Select a chart type:’ Before this point in the interaction the user may not have had an explicit type of chart in mind. The user may not even have known about some of the possibilities. Hence the action taken is an emergent property of the interaction of external and internal resources. Choosing an appropriate chart itself requires an interaction strategy. The gallery of images supports a goal matching strategy or some form of elimination including trial and error. Whatever strategy is chosen, it is embedded within the higher level plan-following strategy for completing the top-level goal.

Step 5 onwards

The process of chart construction continues in much the same way; the user is guided in the selection of parameters and options by the pre-planned sequence of dialogue boxes. In the final steps of this episode, the current state of the chart is represented as an image of how it will look on completion of the dialogue sequence (see Figure 11). This *image of achievement*, in conjunction with the user's internal goal to produce a graphical representation of the original data, can be used to evaluate the quality of the finished product before it is finally produced. It might appear that the user could finish the dialogue here. Whether using a goal matching or a plan-following strategy, the image of achievement provides clear feedback that the goal has been achieved. However, clicking on *Next* produces a further dialogue box, creating further action possibilities, namely to add labels and legends to the chart. This is a continuation of the pre-computed plan that is embedded in the Chart Wizard dialogue.

<Figure 11 about here>

Four key issues emerge from the interaction analysis:

i) The first two steps are poorly supported by external resources. Novice users will not have a plan internalised and will rely heavily on the action-effect resource at the bottom of the screen. This is easy to overlook and can be difficult to interpret. A goal matching strategy will be the only sensible strategy for a novice user in this early part of the interaction. Errors or breakdowns may occur if the user cannot utilise the action-effect information at the bottom of the screen.

ii) After the first two steps the action is shaped by an externalised plan. Unlike the procedures of Figure 4a, the plan that is available is not explicitly stated as a sequence of items. Rather it is implicitly expressed in the form and structure of the dialogue. The user is given feedback about the length of the plan, and her progress through it, but in order to find what the plan is the user would have to navigate forward explicitly through the plan using the appropriate buttons. Similarly, in order to determine what she has done so far, the user must recall the history or explicitly navigate back through the plan using appropriate buttons. We would not expect novice users to have problems with this aspect of steps 3 onwards. We would predict however that at step 2, data selection errors may occur because no history of interaction is externalised.

iii) During this plan-based phase, new sub-goals are externalised, by forcing the user to make new choices and decisions (for instance, by presenting galleries of images showing possible chart types). These trigger possible choices that the user may not have been aware of in advance. Hence the action taken by the user is an emergent property of the interaction between internal and external resources. The user's top-level goal of creating a chart is given a rendering in the interface as an image of the final product before it is actually produced. This instantiation of the user's goal serves as a resource for early evaluation of the whole interaction.

iv) Externalising the plan resource as a sequence of dialogue boxes may make it difficult for the user to internalise the plan (at least in any direct way) because it is implicit. In addition, it does not support the development of alternative interaction strategies because the plan as externalised does not *resource* the interaction. Rather, it *drives* it in the sense of enforcing a plan following strategy. That is to say control structure and information resource are confounded in this particular design solution. Such confounding limits users in terms of what they can do and how they can develop and adapt their strategies.

Our analysis of the dialogue wizard using the resources model allows us to identify a number of characteristics of the design which enhance usability, and also some potential weaknesses in the resources provided by the designers of the Chart Wizard. This is particularly apparent in the early stages of the interaction. The value of the resources model for analysing interaction in this way is not that it makes predictions that other approaches would not. Indeed, a display-based model such as that of Kitajima and Polson (1995) might well predict some of the difficulties in the early steps of the interaction episode. But compared to the connectionist model of Kitajima and Polson, the resources model provides a high-level way of looking at interaction that is easier to apply. Similarly, an evaluation technique such as Cognitive Walkthrough (Wharton, Rieman, Lewis, & Polson, 1994) or Heuristic Evaluation (Nielsen & Molich, 1990) might highlight similar usability problems concerned with lack of screen feedback. But the resources model uses the concepts of interaction strategy and information resources to analyse individual actions and specific display items at a more generic level.

3.3. Interface Design, Resources and Interaction Strategies

In the previous sections we have been concerned to compare and analyse interaction with existing systems. In this section we describe how the resources model can be used to generate different interfaces for the same underlying task and explore how some of these differences affect the interaction strategies users adopt to complete that task. We have chosen to use a simple problem solving task because we wished to compare performance in a more controlled way than would be possible with a lager more complex 'real world' task. The task we have chosen is the 8-puzzle. In its traditional form the 8-puzzle comprises a 3-by-3 square board containing 8 sliding tiles and a space. The tiles are numbered 1 to 8 and the problem solver has to slide the tiles in such a way that she orders the tiles correctly around the outer edge of the board with the tile numbered 1 in the top left hand square and the space in the centre (see figure 12a). Our interest in this puzzle was inspired by recent findings of Golightly and Gilmore (1997) who used it to explore the impact of direct manipulation on interaction strategies using this puzzle.

In the Golightly and Gilmore study participants were asked to solve a computerised version of the *8-puzzle* using two interfaces that differed only in the 'directness' of the interaction that was necessary. In one version, referred to as the direct manipulation (DM) version, the tiles of the puzzle were directly manipulated by clicking on them with a mouse (see Figure 12a). In the second variant, referred to as the *indirect manipulation* (IM) version, tiles were manipulated only indirectly by clicking on a separate array of buttons (Figure 12b, centre panel). In both cases, a fixed display showing the arrangement of pieces in the goal state was also provided (the right hand “Goal” display of Figure 12 (a) and (b)).

<Figure12 about here>

Golightly and Gilmore found that users of the IM interface required *fewer* moves to solve the problem than users of the DM interface. Furthermore, an analysis of the time taken for moves and a qualitative analysis of move patterns suggested that members of the IM group were following a different method from their DM counterparts. The DM users tended to decompose the problem into plans for correctly placing tiles in the first row and in each side but beyond that, they worked on a trial-and-error basis often involving hundreds of moves to achieve each of these sub-goals. We refer to this as the *placement method*.[7] In contrast, IM users planned deeper into the problem, avoiding the trial and error components of the placement method. Golightly and Gilmore did not describe this method in detail but Cockayne (1998) refers to this as the *ring ordering method*. The essence of the ring ordering method is that users first concentrate on getting the tiles in the right order (1-8 in a clockwise) without reference to the absolute positions of the tiles on the board. Only then do they attempt to put the tiles in the correct place on the board. The main focus of planning in this method therefore is to construct a circular array of numbers in the correct order around the outside of the display.

The findings of Golightly and Gilmore are relevant because they demonstrate how different external representations can affect the type of interaction strategy adopted by users. But they were concerned only with exploring designs which change the directness of manipulation. We have used the resources model to generate different versions of the 8-puzzle in a more systematic way. Versions were varied according to which abstract information resources were externalised in the interface (Cockayne 1998; Cockayne, Wright and Fields, 1999).[8] In the traditional puzzle used by Golightly and Gilmore the goal is externalised as a separate fixed display which the user can refer to. We identified at least three ways in which the goal can be externalised in an interface. As with Golightly and Gilmore, the goal state can be externalised as a separate display. Alternatively it can be integrated into the problem solving board as a *background map*. For the traditional version,

---

[7] In fact Golightly and Gilmore refer to this as the placement *strategy*. But to avoid confusion, we will reserve the term strategy to refer to the more general approaches to decision making described in the resources model.

this involves printing the numbers of the correct tile on the display behind the tiles. As a third alternative, the goal can be externalised only through goal achievement feedback. This might involve for example, highlighting the numbers on the tiles only when they are correctly positioned.

We also identified that with the traditional puzzle, the familiar sequence of numbers provides an easy way for the user to internalise the goal, reducing her reliance on the external resources. However, one version of the puzzle was implemented in which colours replaced the numbers. Since the arrangement of colours in the goal state was arbitrary, this proved difficult for the user to represent internally. A further variation used the same colour coding for tiles, with the addition of a background map where the goal state is shown behind the tiles. Now the user has access to an external representation of the goal, but the access is very limited since the tiles obscure most of the information on the background. Studies by Cockayne (1998) show how the different externalisations affect users' interaction strategies.

Our explorations of the eight puzzle show how in a laboratory setting, the resources model can be used to generate a design space for a given task and also how changes to the externalised resources of the 8-puzzle can make coordination of resources more or less difficult. Golightly and Gilmore argued that the directness of manipulation was the significant factor in determining a user's method. Our studies, by using versions that differed only in the resource representations provide stronger evidence that the design of external resources is an important factor in determining interaction strategy.

These are however only laboratory studies of a simple puzzle. Further work on more realistic design problems will be required to determine how the resources model can be effectively used in support of design.

4. DISCUSSION AND CONCLUSIONS

Distributed Cognition makes a valuable contribution to both CSCW and HCI theory and design because it leads us to re-examine the relationship between actors, artefacts and the

---

[8] The work reported here was carried out by Aston Cockayne for a MSc degree under the supervision of the first author.

settings in which interactions occur. By identifying information structures or representations flowing through functional systems as objects of analysis, it becomes possible to reason, not only about designed artefacts, but also cognitive artefacts, within a single conceptual framework.

But the DC research, while identifying resources for action as central to the interaction between people and technologies, stops short of providing a model of such resources. The resources model described in this paper defines a limited number of resource types as abstract information structures. It also identifies a number of interaction strategies that utilise different configurations of resources. These two components of the resources model, information structures and interaction strategies, through the process of co-ordination and integration, provide a link between devices, representations and actions that is not well articulated in the DC literature. We have shown how the resources model can be used to analyse a range of interface applications from checklists of the sort found in aircraft cockpits, through chart wizards, to simple tasks such as the 8-puzzle.

In the spirit of distributed cognition research, the resources model deliberately seeks to soften the boundary between actor and designed artefact. Nardi (1996) has criticised DC research arguing that it places machines and people on an equal epistemic footing. We do not believe that this is an accurate reflection of DC theory (see for example Hutchins & Klausen, 1991) although it is perhaps an understandable misunderstanding. The resources model views both resources and interaction strategies as objects of cognition. That is to say, resources and strategies can be thought about and used in a deliberate and conscious way by human actors. Thus human actors have a different epistemic status to technological artefacts although both are modelled as representational systems. This conception of the relation between human actors, interaction strategies and information resources is thus closer to so-called *interactionist* view of intelligence and problem solving (see Sternberg & Wagner, 1994).

Throughout the paper we have persisted in referring to the resources work as a *model*. This is largely for the sake of convenience since *viewpoint*, *framework* and *analytical approach* are unwieldy terms. But there is a danger here. By using the term model, we may give the impression of completeness and thus leave ourselves open to the charge of leaving something out. For example, why have we not included *mental model* as one of our abstract information types? To counter this we re-iterate that the model may not be complete but is satisfactory for current purposes. We would hope to extend the model as our understanding increases. For example Cockayne, Wright and Fields (1998), explore the idea of *strategy concept* as a resource for action.

A longer term concern is the usefulness of the model in providing designers, evaluators and human factors analysts with a way of describing and reasoning about interaction which is productive for the purpose of design. This kind of validation is difficult to measure in laboratory experiments, case studies or even field studies. It is more a matter of the extent to which a model is taken up by the design community to produce better interfaces. Most HCI models have their strengths and limitations in this regard. The GOMS model is not a complete model. Nevertheless there have been reports of successful application in certain development contexts. With respect to this form of validation, we have supported designers in applying the resources model to the analysis of aircraft failure management interfaces (Hicks, Wright & Pocock 1999). We are also applying the model to the design of virtual environments (Smith, Duke & Wright 1999). This kind of validation takes longer and is an iterative process in which the model will undoubtedly be extended and modified.

One of our hopes for the resources model in the long term is that it may provide a way of relating HCI and CSCW issues more closely. One of our next steps will be to extend the resources model to collaborative settings. We have already applied some aspects of the model to the tasks of air traffic control (Fields, Wright, Marti & Palmonari 1998). This is an interesting application area not only because of the high degree of collaboration between humans in the distributed system, but also because of the salience of plan-following artefacts such as *flight progress strips, stripboards* and *standard approach routes*. Our studies show how, by artful coordination of such resources, components of the complex 4-dimensional problem of maintaining aircraft separation can be reduced to simpler perceptual tasks. In addition our results show that the collaborations between controllers and pilots are mediated by implicit agreements about shared plans and that things do not always go according to the plan. Breakdowns in coordination that result from non-agreed divergences from the plan force controllers and pilots into an explicit representation of the plan (See also Fairburn, Wright & Fields, 1999) and a re-negotiation of the meaning of the plan. In this way plans as resources for action also become artefacts through which power and authority can be exerted. Taking up this political dimension of plans as resources for action we have also explored the status of operating procedures, both as plan resources and artefacts of accountability in the work of commercial aircraft pilots (Wright, Pocock & Fields 1998). Here, we have argued that different communities (pilots, regulatory authorities, designers, etc.) interpret plan-following artefacts such as procedures in ambiguous ways perhaps as resources for action, perhaps as artefacts of accountability. This ambiguity creates a space or context for certain kinds of organisational failure (McCarthy, Wright, Healey & Harrison 1998). More needs to be done to consolidate our framework, but the indications are that the concepts of the resources model can be used as common features in these very different levels of analysis.

NOTES

Background. The work reported here extends previous work by the authors (Wright, Fields & Harrison, 1996; Fields Wright and Harrison, 1997). The studies of the 8-puzzle were carried out by Aston Cockayne as part of his MSc thesis under the supervision of the first author.

Acknowledgements.. Many thanks to John McCarthy at University College Cork and to the members of the York HCI group who have contributed to the development of this paper. Many thanks also to Richard Young, the HCI review editor and three anonymous referees whose comments helped us to present our ideas more clearly.

Support. This work was funded in part by the Esprit *Mefisto* project (Project Number 24963)

Authors' Addresses. Peter Wright, Department of Computer Science, University of York, Heslington, York YO10 5DD, UK. Email peter.wright@cs.york.ac.uk. Robert Fields, School of Computing Science, Middlesex University, Bounds Green Road, London, N11 2NQ, UK. Email b.fields@mdx.ac.uk. Michael, D. Harrison**,** Department of Computer Science, University of York, Heslington, York YO10 5DD, UK. Email michael.harrison@cs.york.ac.uk.

REFERENCES

Card, S.K., Moran, T.P. & Newell, A. (1983) *The psychology of human computer interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Cockayne, A (1998) *Implementing the concept of strategy: Resolving the dilemma of manipulation speed versus learning quality*. Unpublished MSc Thesis, Department of Computer Science, University of York, UK.

Cockayne, A., Wright, P.C. & Fields, B (1999) Supporting interaction strategies through the externalisation of strategy concepts. *Proceedings of INTERACT '99 International Conference on Human-Computer Interaction*., 582-588. Amsterdam: IOF Press.

Degani, A. & Wiener, E.L. (1990) *Human factors of flight-deck checklists: The normal checklist.* (NASA Contractor Report 177549). Moffet Field, CA: NASA-Ames Research Center.

Fairburn, C. Wright, P.C., & Fields, R. (1999) Air traffic control as distributed joint activity: Using Clark's theory of language to understand collaborative work in ATC. *Proceedings of ECCS'99 the European Conference on Cognitive Science*, 21-30. Rome: Istituto di Psicologia Consiglio Nazionale delle Ricerche.

Fields, R.E., Wright, P.C. & Harrison, M.D. (1997) Objectives, strategies and resources as design drivers. *Proceedings of INTERACT '97 International Conference on Human-Computer Interaction*, 164-171. London: Chapman Hall.

Fields, R.E., Wright, P.C., Marti, P. & Palmonari, M. (1998) Air Traffic Control as a Distributed Cognitive System: a study of external representations. *Proceedings of ECCE-9, the 9th European Conference on Cognitive Ergonomics*, 85-90. Roquencourt: EACE Press.

Gibson, J.J. (1966) *The senses considered as perceptual systems.* New York: Houghton Mifflin.

Gibson, J.J. (1979) *The ecological approach to visual perception*. New York: Houghton Mifflin.

Golightly, D. and Gilmore, D. (1997) Breaking the rules of direct manipulation. *Proceedings of INTERACT '97 International Conference on Human-Computer Interaction*, 156-163. London: Chapman Hall.

Green, T.R.G. and Petre, M. (1996) Usability analysis of visual programming environments: A cognitive dimensions framework. *Journal of Visual Languages and Computing*, 7.2, 131-174.

Green, T. R. G., Schiele, F. and Payne, S. J. (1988) Formalisable models of user knowledge in human-computer interaction. In van der Veer, G.C., Green, T. R. G. and Hoc, J-M. (Eds.),Working with Computers: Theory versus outcome, (pp. 3-46). London: Academic Press.

Grudin, J. (1990) The computer reaches out: The historical continuity of user interface design. *Proceedings of CHI '90 Conference on Human Factors in Computer Systems*, 261-266. New York: ACM

Hicks, M., Wright, P., and Pocock, S. (1999) A distributed cognition perspective on civil aircraft failure management system design. *Engineering Psychology and Cognitive Ergonomics* 3, 101-110.

Howes, A. and Payne, S.J. (1990) Display-based competence: towards user models for menu-driven interfaces. *International Journal of Man-Machine Studies*, 33.6, 637-655.

Hutchins, E. (1995a) How a cockpit remembers its speeds. *Cognitive Science*, 19, 126-289.

Hutchins, E. (1995b) *Cognition in the wild*. Cambridge, Mass: MIT Press.

Hutchins, E., Hollan, J.D. and Norman, D. (1986) Direct manipulation interfaces. In Norman, D.A. & Draper, S. (Eds.), *User centered system design: New perspectives on human-computer interaction* (pp. 87-124). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hutchins, E. and Klausen, T. (1991) Distributed cognition in the cockpit. In Engeström, Y. and Middleton, D. (Eds.) *Cognition and communication at work* (pp. 15-34). Cambridge: Cambridge University Press.

Johnson, P. (1992) *Human-computer interaction: Psychology, task analysis and software engineering*. New York: McGraw Hill.

Kieras, D. E. and Polson, P. G. (1985) An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, 22, 365-394.

Kitajima, M and Polson, P.G. (1995) A comprehension-based model of correct performance and errors in skilled, displayed-based human-computer interaction. *International Journal of Human-Computer Studies,* 43.1, 65-100.

Luger, G.F. (1976) Transfer effects in isomorphic problem situations, *International Journal of Man-Machine Studies*, 8, 411-421.

Marr, D. (1982) *Vision: A computational investigation into the human representation and processing of visual information.* San Francisco: Freeman.

Marti, P. (in press) The choice of the unit of analysis for modelling real work settings. *Journal of Cognition, Technology and Work.*

Mayes, R.T., Draper, S.W., McGregor, A.M. and Oatley, K. (1988) Information flow in a user interface: The effect of experience of and context on the recall of MacWrite screens. In Jones, D.M. and Winder, R. (Eds.) *People and computers IV* (pp. 257-289). Cambridge: Cambridge University Press.

McCarthy, J.C., Wright, P.C., Healey, P.G.T. and Harrison, M.D. (1998) Accountability of work activity in high-consequence work systems: human error in context. *International Journal of Human Computer Studies*, 47.6, 735-766.

Monk, A.F. (1990) Action-effect rules: A technique for evaluating an informal specification against principles. *Behaviour and Information Technology,* 9, 147-155.

Monk, A.F. (1999) Modelling cyclic interaction. *Behaviour & Information Technology,* 18.2, 127-139.

Nardi, B.A. (1996) *Context and consciousness: Activity theory and human-computer interaction*. Cambridge, Mass: MIT Press.

Nielsen, J. & Molich, R. (1990) Heuristic evaluation of user interfaces. *Proceedings of the CHI '90 Conference on Human Factors in Computer Systems,* 249-256. New York: ACM

Norman, D.A. (1986) Cognitive engineering. In Norman, D.A. and Draper, S. (Eds.) *User centered system design: New perspectives on human-computer interaction* (pp. 31-61 Hillsdale, NJ: Lawrence Erlbaum Associates.

Norman, D.A. (1988). *The psychology of everyday things*. New York: Basic books.

Norman, D.A. (1999) Affordances, conventions and design. *Interactions,* VI.3, 38-43.

Payne, S.J. & Green, T.R.G., (1986) Task-action Grammars: A model of the mental representation of task languages. *Human Computer Interaction*, 2, 93-133.

Rogers, Y. and Ellis, J. (1994) Distributed Cognition: an alternative framework for analysing and explaining collaborative working. *Journal of Information Technology*, 9.2, 119-128.

Russell. S.J. and Norvig, P. (1995) *Artificial intelligence: A modern approach*. Englewood Cliffs: Prentice Hall.

Scaife, M. and Rogers, Y. (1996), External cognition: How do graphical representations work? *International Journal of Human-Computer Studies*, 45, 185-213.

Schmidt, K. (1997) Of maps and scripts: The status of formal constructs in cooperative work. *Proceedings of the Group'97 International Conference on Supporting Group Work*, 138-147. New York: ACM.

Simon, H.A. (1975) The functional equivalence of problem solving skills. *Cognitive Psychology*, 7, 268-272.

Smith, S., Duke, D. and Wright, P.C. (1999) Using the resources model in virtual environment design. *King's Manor Workshop on User-Centred Design and Implementation of Virtual Environments.* University of York.

Sternberg, R.J. and Wagner, R.K. (1994) *Mind in Context: Interactionist perspectives on human intelligence*. Cambridge: Cambridge University Press.

Suchman, L.A. (1987) *Plans and situated actions: The problem of human computer interaction*. Cambridge: Cambridge University Press.

Suchman, L.A. (1993) Response to Vera and Simon's Situated action: A symbolic interpretation. *Cognitive Science,* 17, 71-77.

Wharton, C., Rieman, J., Lewis, C and Polson, P. (1994) The cognitive walkthrough method: A practitioner's guide. In Nielsen, J., and Mack, R. (Eds.) *Usability inspection methods* (pp. 105-140). New York: Wiley.

Wright, P.C., Fields, R.E. and Harrison, M.D. (1996) Distributed information resources: A new approach to interaction modelling. *Proceedings of ECCE-8, the 8th European Conference on Cognitive Ergonomics,* 5-10. Roquencourt: EACE Press.

Wright, P.C., Fields, R.E. and Merriam, N.A. (1997) From formal models to empirical evaluation and back again. In Palanque, P., and Paternò, F. (Eds.) *Formal methods in human-computer interaction* (pp. 283-314). Berlin: Springer.

Wright, P., Pocock, S., and Fields, R.E. (1998) The prescription and practice of work on the flight deck. *Proceedings of ECCE-8, the 8th European Conference on Cognitive Ergonomics*, 37-42. Roquencourt: EACE Press.

Young, R.M., Howes, A. and Whittington J. (1990) A knowledge analysis of interactivity. *Proceedings of INTERACT '90 International Conference on Human-Computer Interaction*, 115-120. Amsterdam: North Holland.

Zhang, J. (1996). A representational analysis of relational information displays. *International Journal of Human-Computer Studies*, 45, 59-74.

Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive Science*, 21.2, 179-217.

Zhang, J. and Norman, D.A. (1994) Representations in distributed cognitive tasks. *Cognitive Science,* 18, 87-122.

Figure captions

Figure 1: The cycle of interaction in the resources model.

Figure 2: Strategies and the resources they require.

Figure 3: Three different representations of goal and state for an airspeed indicator.

Figure 4: Three different representations of plan and history for a procedure display.

Figure 5: Charting making using a plan-based interface.

Figure 6: Charting making using a dialogue box.

Figure 7: The spreadsheet screen at the start of the interaction episode.

Figure 8: The screen after clicking on the Chart Wizard button (start of step 2).

Figure 9: The screen after dragging a presentation area (start of step 3).

Figure 10: The screen after selecting data (start of step 4).

Figure 11: A dialogue box showing an image of the finished product.

Figure 12: Direct (a) and indirect (b) manipulation user interfaces for the 8 puzzle. In (a) the user clicks directly on tiles in the left panel to make a move. In (b) the user must click on buttons in the centre panel to move tiles in the left panel.

Figures

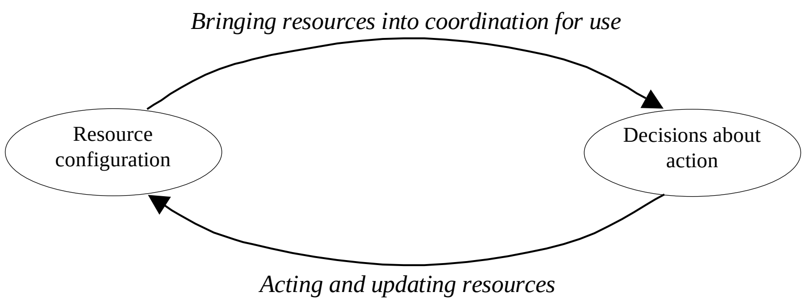
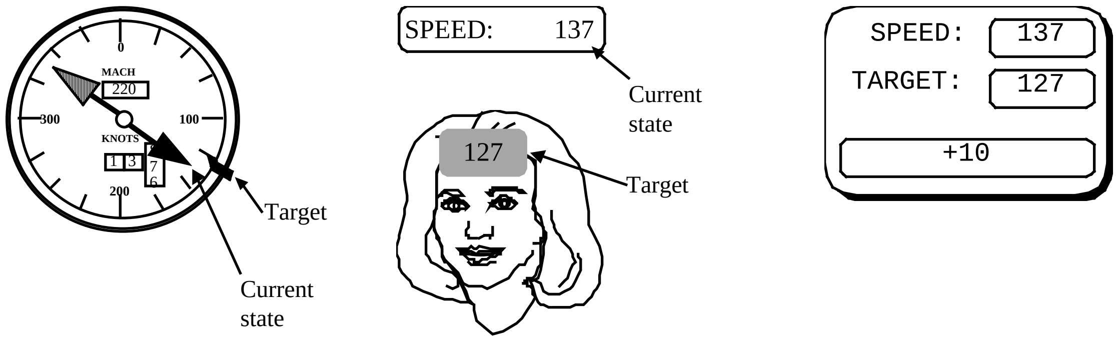


Figure 1

Strategy	**Resources required**
plan following	plan, history state and goal
plan construction	goal, possibilities, action-effects and state
goal matching	goal, possibilities, action-effects
history-based choice	goal, possibilities, history

Figure 2.


0
MACH
220
300
100
KNOTS
1 3
7
6
200
Target
Current state
SPEED: 137
Current state
127
Target
SPEED: 137
TARGET: 127
+10

(a) *(b)* *(c)*

Figure 3

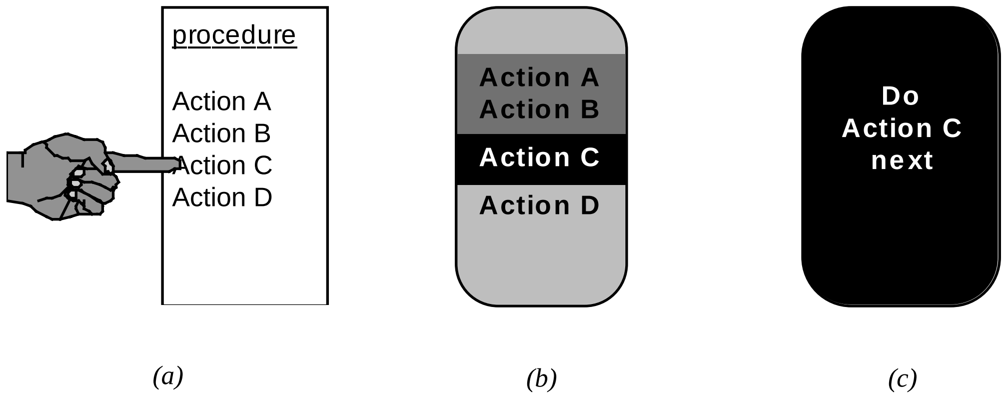
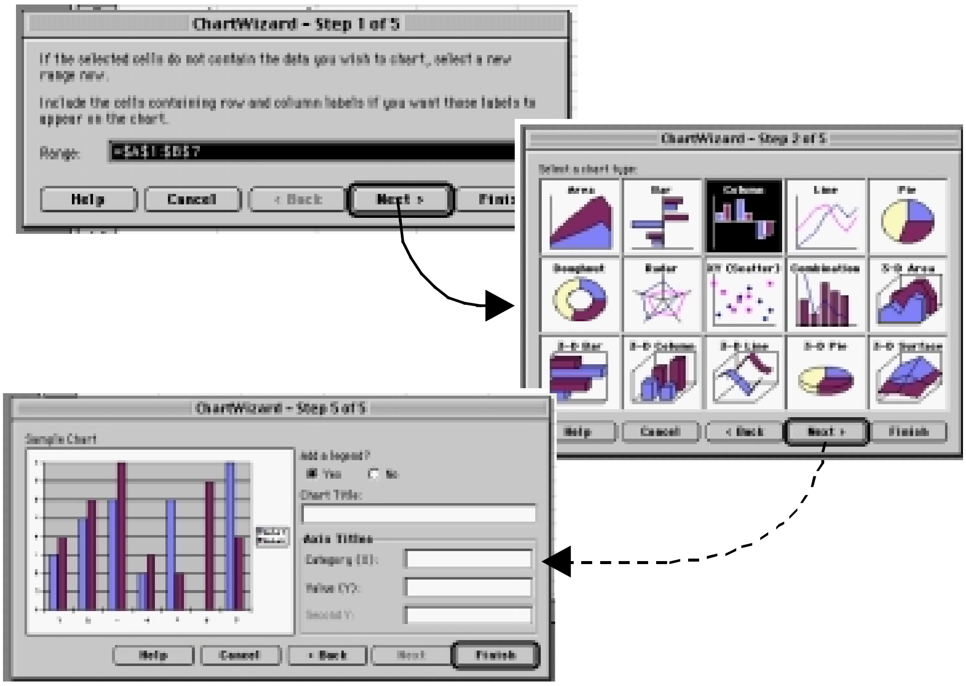

procedure
Action A
Action B
Action C
Action D
Action A
Action B
Action C
Action D
Do
Action C
next
(a)
(b)
(c)

Figure 4



ChartWizard - Step 1 of 5
If the selected cells do not contain the data you wish to chart, select a new range now.
Include the cells containing row and column labels if you want those labels to appear on the chart.
Range:
Help
Cancel
< Back
Next >
ChartWizard - Step 2 of 5
Select a chart type:
Area
Bar
Column
Line
Pie
Doughnut
Radar
XY (Scatter)
Combination
3-D Area
3-D Bar
3-D Column
3-D Line
3-D Pie
3-D Surface
Help
Cancel
< Back
Next >
Finish
ChartWizard - Step 5 of 5
Sample Chart
Add a legend?
Yes
No
Chart Title:
Axis Titles
Category (X):
Value (Y):
Second Y:
Help
Cancel
< Back
Next
Finish

Figure 5

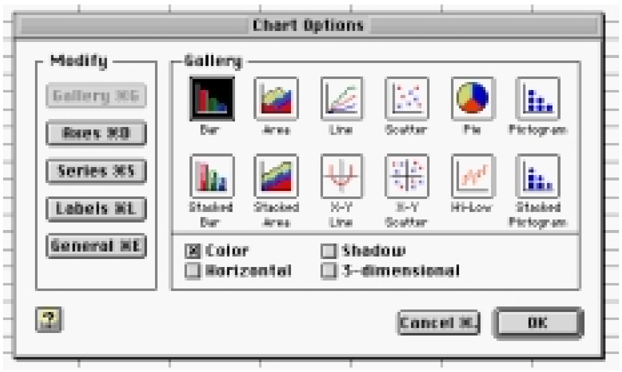


Chart Options
Modify
Gallery ⌘G
Axes ⌘A
Series ⌘S
Labels ⌘L
General ⌘E
Gallery
Bar
Area
Line
Scatter
Pie
Pictogram
Stacked Bar
Stacked Area
X-Y Line
X-Y Scatter
Hi-Low
Stacked Pictogram
Color
Shadow
Horizontal
3-dimensional
Cancel
OK

Figure 6

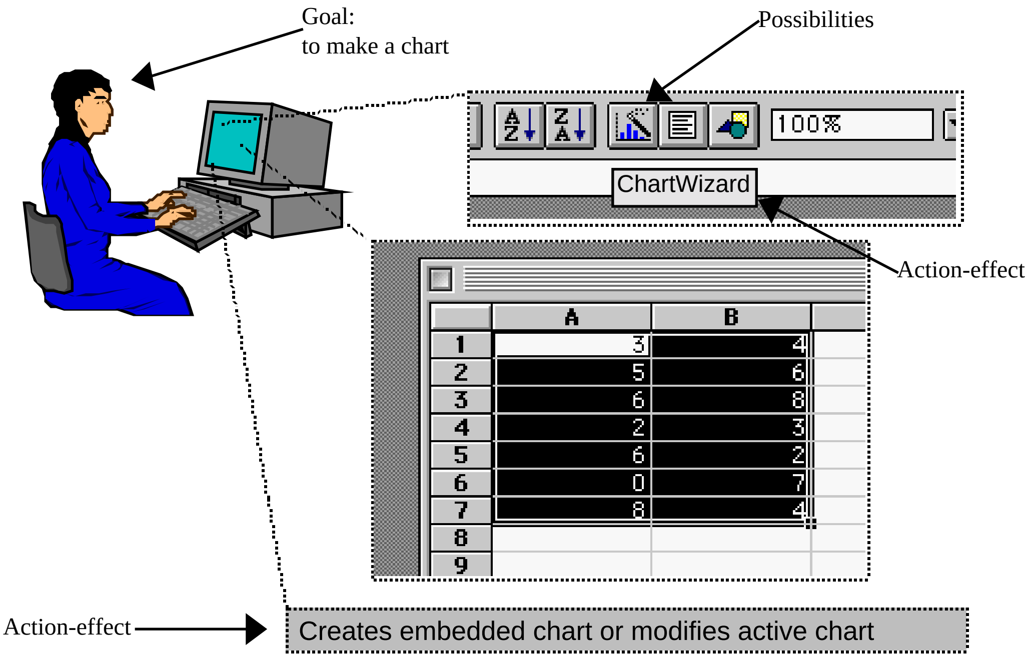

Goal:
to make a chart
Possibilities
100%
ChartWizard
Action-effect
A
B
1
3
4
2
5
6
3
6
8
4
2
3
5
6
2
6
0
7
7
8
4
8
9
Action-effect
Creates embedded chart or modifies active chart

Figure 7

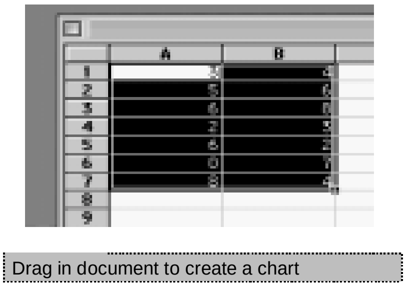


Figure 8

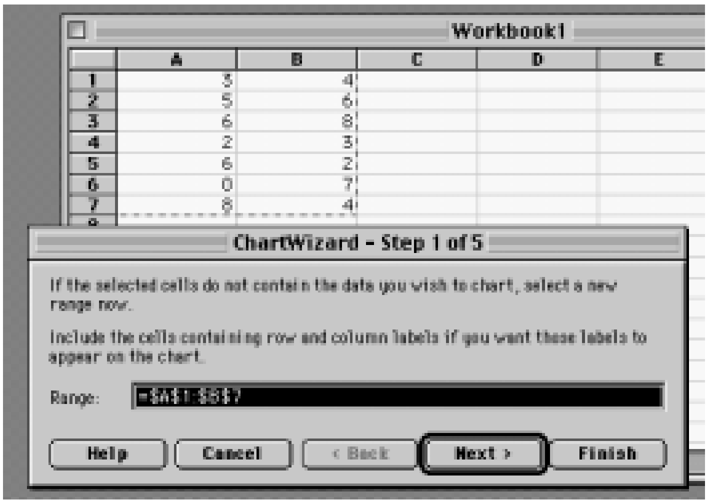

Workbook1
A
B
C
D
E
ChartWizard - Step 1 of 5
If the selected cells do not contain the data you wish to chart, select a new range now.
Include the cells containing row and column labels if you want those labels to appear on the chart.
Range:
Help
Cancel
< Back
Next >
Finish

Figure 9

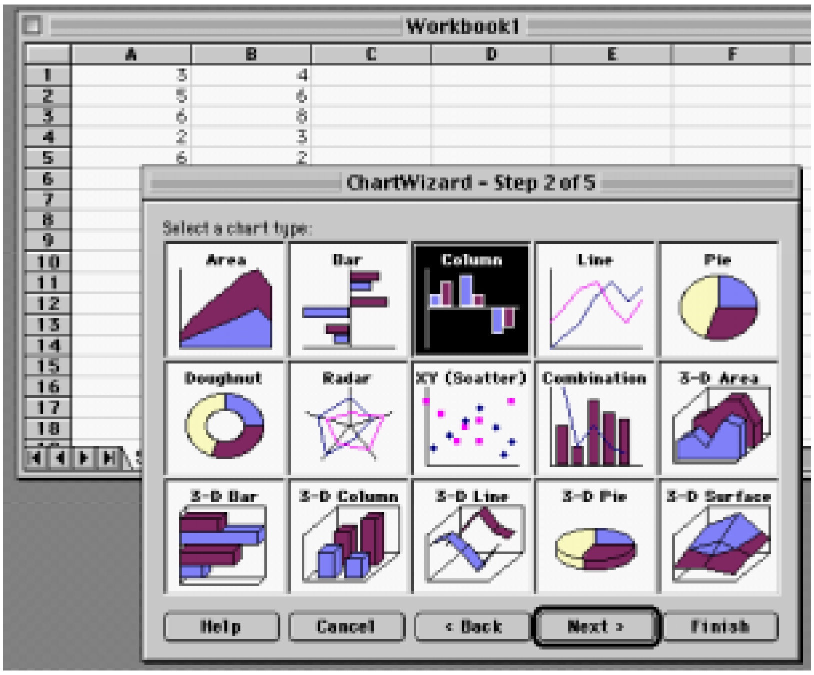

Workbook1
A
B
C
D
E
F
ChartWizard - Step 2 of 5
Select a chart type:
Area
Bar
Column
Line
Pie
Doughnut
Radar
XY (Scatter)
Combination
3-D Area
3-D Bar
3-D Column
3-D Line
3-D Pie
3-D Surface
Help
Cancel
< Back
Next >
Finish

Figure 10

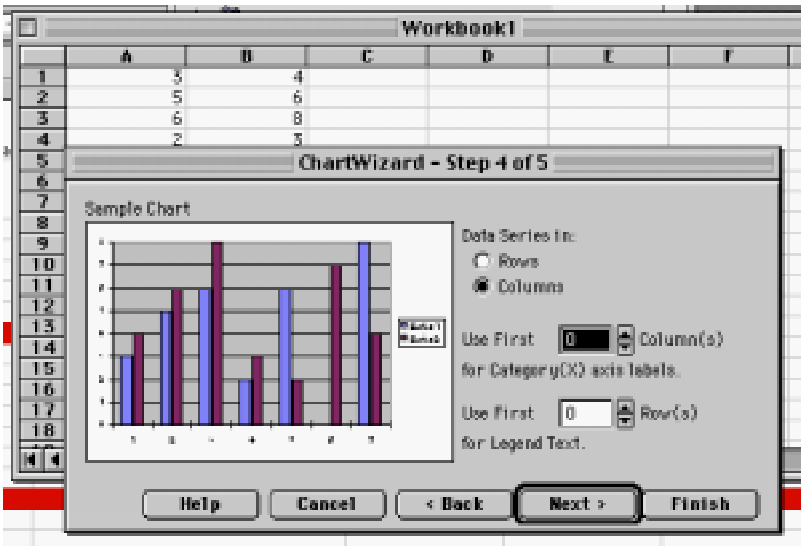

Workbook1
ChartWizard - Step 4 of 5
Sample Chart
Data Series in:
Rows
Columns
Use First
Column(s)
for Category(X) axis labels.
Use First
Row(s)
for Legend Text.
Help
Cancel
< Back
Next >
Finish

Figure 11

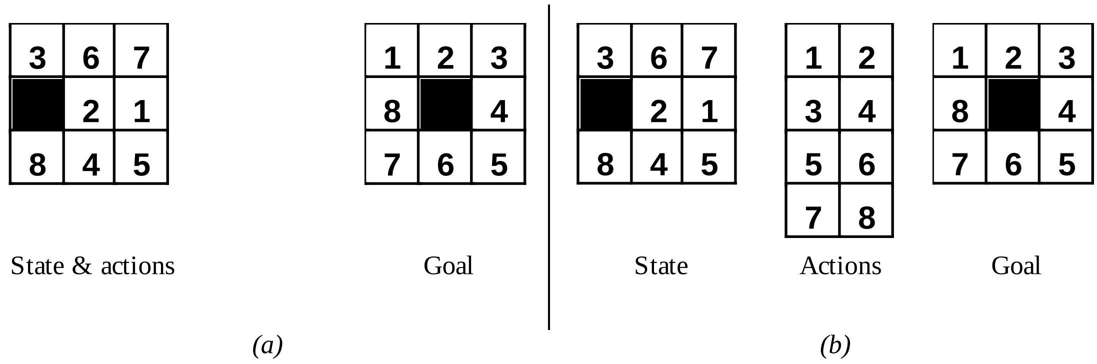

3 6 7
2 1
8 4 5
State & actions
1 2 3
8 4
7 6 5
Goal
(a)
3 6 7
2 1
8 4 5
State
1 2
3 4
5 6
7 8
Actions
1 2 3
8 4
7 6 5
Goal
(b)

Figure 12.