Accepted for publication in E R G O N O M I C S , 1 99 8 , vol 4 1 , 2, 1 2 6 -1 3 9

Conception of the Cognitive Engineering design problem

John Dowell† and John Long‡

†Centre for HCI Design, City University, Northampton Square, London. EC1V 0HB
‡Ergonomics and HCI Unit, University College London, 26 Bedford Way, London. WC1H 0AP, UK.

Cognitive design, as the design of cognitive work and cognitive tools, is predominantly a craft practice which currently depends on the experience and insight of the designer. However the emergence of a discipline of Cognitive Engineering promises a more effective alternative practice, one which turns on the prescription of solutions to cognitive design problems. In this paper, we first examine the requirements for advancing Cognitive Engineering as a discipline. In particular, we identify the need for a conception which would provide the concepts necessary for explicitly formulating cognitive design problems. A proposal for such a conception is then presented.

1. Discipline of Cognitive Engineering

1.1. Evolution of Cognitive Design

A recurrent assumption about technological progress is that it derives from, or is propelled by, the application of scientific theory. Design is seen principally as an activity which translates scientific theory into useful artifacts. As such, design does not possess its own knowledge, other than perhaps as the derivative of a purer scientific knowledge. Yet close examination (Layton, 1974; Vincenti, 1993) shows this view to be in contradiction of the facts. The more correct analysis suggests that technology disciplines acquire and develop their own knowledge which enables them to solve their design problems (Long and Dowell, 1996).

The analysis of "technology as knowledge" (Layton, 1974) recognises the variety of forms of technological knowledge, ranging from tacit 'know how' and 'know what', based on personal experience, to validated engineering principles. Consider the evolution of a new technology. New technologies invariably emerge from the “inspired tinkering” (Landes, 1969) of a few who see a direct route between innovation and exploitation. As an industry is established, *ad hoc* innovation is supplanted by more methodical practices through which the experience of prior problems is codified and re-used. Design is institutionalised as a craft discipline which supports the cumulation and sharing of techniques and lessons learnt. The knowledge accumulated is only marginally, or indirectly derivative of scientific

theory. In the case of computing technology, for example, Shaw has observed: " Computer science has contributed some relevant theory but practice proceeds largely independently of this organised knowledge (Shaw, 1990)".

This same observation can be made of cognitive design, the activity of designing cognitive work and cognitive tools (including interactive computational tools). To date, the seminal successes in cognitive design have been principally the result of inspired innovation. The graphical user interface arose from the careful application of experience cast as design heuristics, for example, “Communicate through metaphors” (Johnson, Roberts, Verplank, Irby, Beard and Mackey, 1989). The spreadsheet is another example. More recent advances in "cognitive technologies", such as those in groupware, dynamic visualisation techniques, and multimedia, are no different in arising essentially through craft practice based on innovation, experience and prior developments. Nevertheless, in the wake of these advances, a craft discipline has been established which supports the cumulation and sharing of knowledge of cognitive design.

However the history of technological disciplines also indicates that continued progress depends on the evolution of a corpus of validated theory to support them (Hoare, 1981; Shaw, 1990). Craft disciplines give way to engineering disciplines: personal experiential knowledge is replaced by design principles; 'invent and test' practices (that is to say, trial-and-error) are replaced by 'specify then implement' practices. Critically, design principles appear not to be acquired by translation of scientific theories. Rather, they are developed through the validation of knowledge about design problems and how to solve them.

The evolution of an engineering discipline is a visible requirement for progress in cognitive design. The requirement is apparent in at least three respects. First, cognitive design needs to improve its integration in systems development practices, and to ensure it has a greater influence in the early development life of products. Second, cognitive design needs to improve the reliability of its contributions to design, providing a greater assurance of the effectiveness of cognitive work and tools. Third, cognitive design needs to improve its learning process so that knowledge accumulated in successful designs can be made available to support solutions to new design problems. For at least these reasons, cognitive design must advance towards an engineering discipline. This paper is addressed to the evolution of such a discipline, a discipline of Cognitive Engineering.

1.2. Emergence of Cognitive Engineering

The idea of a discipline of Cognitive Engineering has been advocated consistently for more than a decade (Hollnagel and Woods, 1983; Norman, 1986; Rasmussen, Pejtersen, and Goodstein, 1994; Woods, 1994). Norman has described Cognitive Engineering as a discipline which has yet to be constructed but whose promise is to transform cognitive design by supplying the "principles that get the design to a pretty good state the first time around (Norman, 1986)". The aims of Cognitive Engineering are "to understand the fundamental principles behind human action and performance that are relevant for the development of engineering principles of design", and second, "to devise systems that are pleasant to use". The critical phenomena of Cognitive Engineering include tasks, user action, user conceptual models and system image. The critical methods of Cognitive Engineering include approximation, and treating design as a series of trade-offs including giving different priorities to design decisions (Norman, 1986).

Woods (1994) describes Cognitive Engineering as an approach to the interaction and cooperation of people and technology. Significantly, it is not to be taken as an applied Cognitive Science, seeking to apply computational theories of mind to the design of systems of cognitive work. Rather, Cognitive Engineering is challenged to develop its own theoretical base. Further, "Cognitive systems are distributed over multiple agents, both people and machines" (Woods, 1994) which cooperatively perform cognitive work. Hence the unit of analysis must be the joint and distributed cognitive system. The question which Cognitive Engineering addresses is how to maximise the overall performance of this joint system. Woods and Roth (1988) state that this question is not answered simply through amassing ever more powerful technology; they contrast such a technology-driven approach with a problem-driven approach wherein the "requirements and bottlenecks in cognitive task performance drive the development of tools to support the human problem solver". Yet whether such an approach may be developed remains an open question: whether designers might be provided with the "concepts and techniques to determine what will be useful, or are we condemned to simply build what can be practically built and wait for the judgement of experience?" Woods and Roth re-state this as ultimately a question of whether "principle-driven design is possible".

1.3. Discipline matrix of Cognitive Engineering

Cognitive Engineering is clearly an emerging discipline whose nucleus has been in research aiming to support cognitive design. The breadth and variety of its activity has continued to grow from its inception and the question now arises as to how the evolution of this discipline can be channelled and hastened. It is here that reference to Kuhn's analysis of paradigms in the (physical and biological) sciences may offer guidance (Kuhn, 1970). Specifically, Kuhn identifies the principal elements of a 'discipline matrix' by which a discipline emerges and evolves. We might similarly interpret the necessary elements of the 'discipline matrix' of Cognitive Engineering.

The first element described by Kuhn is a "shared commitment to models" which enables a discipline to recognise its scope, or ontology. (Kuhn gives the example of a commitment to the model of heat conceived as the kinetic energy of the constituent parts of masses). For Cognitive Engineering, we may interpret this requirement as the need to acquire a conception of the nature and scope of cognitive design problems. Similarly, as Carroll and Campbell have argued, "the appropriate ontology, the right problems and the right ways of looking at them ... have to be in place for hard science to develop (Carroll and Campbell, 1986)". Features of a conception for Cognitive Engineering are already apparent, for example, in Wood's assertion that the unit of analysis must be the distributed cognitive system

A second element of the disciplinary matrix is "values" which guide the solution to problems. Kuhn gives the example of the importance which science attaches to prediction. Cognitive Engineering also needs to establish its values, an example is the value attached to design prescription: "(getting) the design to a pretty good state the first time around (Norman, 1986)"

A third element is "symbolic generalisations" which function both as laws and definitions for solving problems. Kuhn gives the example of Ohm's Law which specifies the lawful relationships between the concepts of resistance, current and voltage. For Cognitive Engineering, we may interpret this requirement as the need for engineering principles which express the relations between concepts and which enable design prescription. The need for engineering principles is one which has been recognised by both Norman and by Woods.

The final element of the disciplinary matrix is "exemplars" which are instances of problems and their solutions. Exemplars work by exemplifying the use of models, values and symbolic generalisations, and they support reasoning about similarity relations with new and unsolved problems. Kuhn gives the example of the application of Newton's second law to predicting the motion of the simple pendulum. (Note, Newton's second law embodies the concept of inertia established in the model of mechanics which commences the *Principia*). Cognitive Engineering too must acquire exemplars, but here those exemplars are instances of solutions to cognitive design problems, together with the design practices which produced those solutions. Such design exemplars must illustrate the application of the conception, values and design principles and must allow designers to view new cognitive design problems as similar to problems already solved.

1.4. Requirements for a conception

If this analysis of the discipline matrix of Cognitive Engineering is correct, then it is also apparent that the necessary elements substantially remain to be constructed. None are particularly apparent in the craft-like discipline of Human Factors which, for example, does not possess engineering principles, the heuristics it possesses being either 'rules of thumb' derived from experience or guidelines derived informally from psychological theories and findings.

This paper is concerned with the requirement for a conception of cognitive design. As later explained, we believe this is the element of the Cognitive Engineering matrix which can and should be established first. The current absence of a conception of cognitive design is well recognised; for example, Barnard and Harrison (1989) called for an "integrating framework .... that situates action in the context of work .... and relates system states to cognitive states", a call which still remains unanswered. However it would be wrong to suggest that currently there is no available conception of cognitive design. Rather, there are many alternative and conflicting conceptions, most being informal and partial. Hollnagel (1991) was able to characterise three broad kinds of conception: the computer as 'interlocutor', with cognitive work seen as a form of conversation with cognitive tools; the "human centred" conception, wherein cognitive work is understood in terms of the user's experience of the world and its mediation by tools; and the 'systems understanding' in which the worker and tools constitute a socio-technical system acting in a world. The last form of conception most clearly conforms with Woods' requirements for Cognitive Engineering, as detailed above.

Previously we have proposed a conception of the cognitive design problem (Dowell and Long, 1989; see also, Long and Dowell, 1989) intended to contribute to the discipline matrix of Cognitive Engineering. That proposal is re-stated in revised form below.

2 Conception of the Cognitive Engineering design problem

Cognitive design concerns the problems of designing effective cognitive work, and the tools with which we perform that work. Our conception of the general problem of Cognitive Engineering is formulated over concepts of cognitive work and tools, and the need to prescribe effective solutions to the cognitive design problems they

present. The concepts are highlighted on first reference. A glossary appears at the end of the paper.

Cognitive work is performed by *worksystems* which use knowledge to produce intended changes in environments, or *domains*.Worksystems consist of both human activity and the tools which are used in that activity (Mumford, 1995). Domains are organised around specific goals and contain both possibilities and constraints. For example, the domain of Air Traffic Management is defined by the goals of getting aircraft to their destinations safely, on time, and with a minimum of fuel use, etc. This domain has possibilities, such as vacant flight levels and the climbing abilities of different aircraft; it also has constraints, such as rules about the legal separation of aircraft. Cognitive work occurs when a particular worksystem uses knowledge to intentionally realise the possibilities in a particular domain to achieve goals. The air traffic controllers, for example, use their knowledge of individual flights, and of standard routes through some airspace, to instruct aircraft to maintain separations and best flight tracks. In this way, the controllers act intentionally to provide a desired level of safety and 'expedition' to all air traffic.

Cognitive tools support the use of knowledge in cognitive work. Those tools provide representations of domains, processes for transforming those representations, and a means of expressing those transformations in the domains (Simon, 1969). The radar and other devices in the Air Traffic Controller's suite, for example, provide representations which enable the controller to reason about the state of the domain, such as aircraft proximities, and to transform those representations, including issuing instructions to pilots, so expressing the controller's activity in the air traffic management domain. The controller's tools embed the intention of their designers of helping the controller achieve their goals. In spite of the way we may often casually describe what we are doing, it is never the case that the our real intention is one of using a tool. Rather, our intention is to do ‘something’ with the tool. The difficulty we have, in describing exactly what that something is, stems from the fact that the domains in which we perform cognitive work are often virtual worlds, far removed from physical objects (for instance, computer-mediated foreign exchange dealing).

The worksystem clearly forms a dualism with its domain: it therefore makes no sense to consider one in isolation of the other (Neisser, 1987). If the worksystem is well adapted to its domain, it will reflect the goals, regularities and complexities in the domain (Simon, 1969). It follows that the fundamental unit of analysis in cognitive design must be the worksystem whose agents are joined by the common intention of performing work in the domain (see also Rasmussen and Vicente, 1990; Woods, 1994). Within the worksystem, human activity is said to be intentional, the behaviour of tools is said to be intended.

The following sections outline a conception of cognitive work informed by systems design theory (e.g., Simon, 1969; Checkland, 1981), ecological systems theory (e.g., Neisser, 1987), cognitive science (e.g., Winograd and Flores, 1986) and Cognitive Engineering theory (e.g., Woods, 1994). It provides a related set of concepts of the worksystem as a system of human and device agents which use knowledge to perform work in a domain.

2.1 Domains of cognitive work

The domains of cognitive work are abstractions of the 'real world' which describe the goals, possibilities and constraints of the environment of the worksystem. Beltracchi (1987, see Rasmussen and Vicente (1990)), for example, used the Rankine

Cycle to describe the environment of process controllers. However, for most domains, such formal models and theories are not available, even for ubiquitous domains such as document production. Further too, such theories do not provide explicit or complete abstractions of the goals, possibilities and constraints for the decision-making worksystem. For example, the Rankine cycle leaves implicit the goal of optimising energy production (and the sub-goals of cycle efficiency, etc), and is incomplete with regard to the variables of the process (e.g., compressor pressure) which might be modified. The conception must therefore provide concepts for expressing the goals, possibilities and constraints for particular instances of domains of cognitive work.

Domains can be conceptualised in terms of *objects* identified by their *attributes*. Attributes emerge at different levels within a hierarchy of complexity within which they are related (energy cycle efficiency and feedwater temperature, for one example, or the safety of a set of air traffic and the separations of individual aircraft for another example). Attributes have *states* (or values) and may exhibit the *affordance* for change. Desirable states of attributes we recognise as *goals*, for instance, specific separations between aircraft, and specific levels of safety of air traffic being managed. *Work* occurs when the attribute states of objects are changed by the behaviours of a worksystem whose intention it is to achieve goals. However work does not always result in all goals being achieved all of the time, and the difference between the goals and the actual state changes achieved are expressed as *task quality* .

The worksystem has a *boundary* enclosing all user and device behaviours whose intention is to achieve the same goals in a given domain. Critically, it is only by defining the domain that the boundary of the worksystem can be established: users may exhibit many contiguous behaviours, and only by specifying the domain of concern, might the boundary of the worksystem enclosing all relevant behaviours be correctly identified. Hence, the boundary may enclose the behaviours of more than one device as, for example, when a user is working simultaneously with electronic mail and bibliographic services provided over a network. By the same token, the worksystem boundary may also include more than one user as, for example, in the case of the air traffic controller and the control chief making decisions with the same radar displays.

The centrality of the task domain has not always been accepted by cognitive design for research, with significant theoretical consequences. Consider the GOMS model (Card, Moran and Newell, 1983). Within this model, goals refer to states of "the user's cognitive structure" referenced to the user interface; actions (methods) are lower level decompositions of goals. Hence a seminal theory in cognitive design leaves us unable to distinguish *in kind* between our goals and the behaviours by which we seek to achieve those goals.

2.2 Worksystem as cognitive structures and behaviours

Worksystems have both structures and behaviours. The *structures* of the worksystem are its component capabilities which, through coupling with the domain, give rise to behaviour. *Behaviours* are the activation (see Just and Carpenter, 1992; also Hoc, 1990) of structures and ultimately produce the changes in a domain which we recognise as work being performed.

Consider the structures and behaviours of a text editor. A text editor is a computer for writing, reading and storing text. Text is a domain object and is both real and virtual. At a low level of description, usually invisible to the user, text appears as

data files stored in a distinct code. At a higher level, text consists of information and knowledge stored in a format which the user may choose. Text objects have attributes, such as character fonts at one extreme and the quality of prose at the other. Generally, the domain is represented by the text editor only partially and only at low and intermediate levels. The program is a set of structures, including functions, such as formatting commands, as well as menus, icons and windows. In simple text editors, the program is a fixed invariant structure; more sophisticated editors allow the user to modify the structure - users can choose which functions are included in the program, which are presented on the menus, and the parameters of the processes they specify. These structures are activated in the behaviours of the text editor when text is created, revised and stored. Higher level editor behaviours would include browsing and creating tables of contents through interaction with the user. With these behaviours, text which has themes, style and grammar is created by users.

As this example indicates, structures consist of *representations* (e.g., for storing text) and *processes* (e.g., text editing processes). Behaviours (e.g., creating and editing text) are exhibited through activating structures when processes (e.g., functions) transform representations (e.g., text). Behaviours are the processing of representations.

2.3 Cognitive structures and behaviours of the user

Users too can be conceptualised in terms of structures and behaviours by limiting our concern for the person to a cognitive agent performing work. The user's cognitive behaviours are the processing of representations. So, perception is a process where a representation of the domain, often mediated by tools, is created. Reasoning is a process where one representation is transformed into another representation. Each successive transformation is accomplished by a process that operates on associated representations. The user's cognitive behaviours are both abstract (i.e., mental) and physical. Mental behaviours include perceiving, knowing, reasoning and remembering; physical behaviours include looking and acting. So, the physical behaviour of looking might have entailed the mental behaviours of reasoning and remembering, that is why and where to look. These behaviours are related whereby mental behaviours generally determine, and are expressed by, the user's physical behaviours. A user similarly possesses cognitive structures, an architecture of processes and representations containing knowledge of the domain and of the worksystem, including the tools and other agents with which the user interacts.

Propositions, schema, mental models and images are all proposals for the morphology of representations of knowledge. The organisation of the memory system, associative and inductive mechanisms of learning, and constraints on how information can be represented (such as innate grammatical principles) have all been proposed as aspects of cognition and its structural substrates.

However, such theories established in Cognitive Science may not, in fact, have any direct relevance for the user models needed for designing cognitive work. To assume otherwise would be to conform with the view of (cognitive) design as an applied (cognitive) science, a view which we rejected at the beginning of this paper. Simply, the computational theory of mind is not concerned with how the symbols manipulated by cognition have a meaning external to the processes of manipulation, and therefore how they are grounded in the goals, constraints and possibilities of task domains (Hutchins, 1994; McShane, Dockrell and Wells, 1992). As a

consequence, it is very likely the case that many theories presented by Cognitive Science to explain the manipulation of symbols cannot themselves be grounded in particular domains (see also Vicente,1990).

It is rather the case that Cognitive Engineering must develop its own models of the user as cognitive agent. In this development, the ecology of user cognition with the domain must be a fundamental assumption, with models of user cognition reflecting the nature of the domains in which cognitive work is performed. Such an assumption underpins the validity of models in Cognitive Engineering: “If we do not have a good account of the information that perceivers are actually using, our hypothetical models of their information processing are almost sure to be wrong. If we do have such an account, however, such models may turn out to be almost unnecessary"(Neisser, 1987).

2.4 Worksystem as hierarchy

The behaviours of the worksystem emerge at hierarchical levels where each level subsumes the underlying levels. For example, searching a bibliographic database for a report subsumes formulating a database query and perhaps iteratively revising the query on the basis of the results obtained. These behaviours themselves subsume recalling features of the report being sought and interpreting the organisation of the database being accessed.

The hierarchy of behaviours ultimately can be divided into abstract and physical levels. Abstract behaviours are generally the extraction, storage, transformation and communication of information. They represent and process information concerning: domain objects and their attributes, attribute relations and attribute states, and goals. Physical behaviours express abstract behaviours through action. Because they support behaviours at many levels, structures must also exist at commensurate levels.

The hierarchy of worksystem behaviours reflects the hierarchy of complexity in the domain. The worksystem must therefore have behaviours at different levels of abstraction equivalent to the levels at which goals are identified in the domain. Hence a complete description of the behaviours of an authoring worksystem, for example, must describe not only the keystroke level behaviours relating to the goal of manipulating characters on a page, but it must also describe the abstract behaviours of composition which relate to the goals of creating prose intended to convey meaning. Traditional task analyses describe normative task performance in terms of temporal sequences of overt user behaviours. Such descriptions cannot capture the variability in the tasks of users who work in complex, open domains. Here, user behaviour will be strongly determined by the initial conditions in the domain and by disturbances from external sources (Vicente, 1990). In complex domains, the same task can be performed with the same degree of effectiveness in quite different ways. Traditional task analyses cannot explain the 'intelligence' in behaviour because they do not have recourse to a description of the abstract and mental behaviours which are expressed in physical behaviours.

The hierarchy of worksystem behaviours is distributed over the agents and tools of the worksystem (i.e., its structures). It is definitional of systems (being 'greater than the sum of their parts') that they are composed from sub-systems where "the several components of any complex system will perform particular sub-functions that contribute to the overall function" (Simon, 1969). The functional relations, or "mutual influence" (Ashby, 1956), between the agents and between the agents and tools of the worksystem are *interactions* between behaviours. These interactions fundamentally

determine the overall worksystem behaviours, rather than the behaviours of individual agents and tools alone. The *user interface* is the combination of structures of agents and tools supporting specific interacting behaviours (see Card, Moran and Newell, 1983). Norman (1986) explains that the technological structures of the user interfaces are changed through design, whilst the user cognitive structures of the user interface are changed through experience and training.

2.5 Costs of cognitive work

Work performed by the worksystem will always incur *resource cost*s which may be structural or behavioural. Structural costs will always occur in providing the structures of the worksystem. Behavioural costs will always occur in using structures to perform work.

Human structural costs are always incurred in learning to perform cognitive work and to use cognitive tools. They are the costs of developing and maintaining the user's knowledge and cognitive skills through education, training and gaining experience. The notion of learnability refers generally to the level of structural resource costs demanded of the user.

Human behavioural costs are always incurred in performing cognitive work. They are both physical and mental. Physical costs are the costs of physical behaviours, for example, the costs of making keystrokes on a keyboard and scrutinising a monitor; they may be generally expressed as physical workload. Mental behavioural costs are the costs of mental behaviours, for example, the costs of knowing, reasoning, and deciding; they may be generally recognised as mental workload. Behavioural cognitive costs are evidenced in fatigue, stress and frustration. The notion of usability refers generally to the level of behavioural resource costs demanded of the user.

2.6 Worksystem performance

The *performance* of a worksystem relates to its achievement of goals, expressed as task quality, and to the resource costs expended. Critically then, the behaviour of the worksystem is distinguished from its performance, in the same way that 'how the system does what it does' can be distinguished from 'how well it does it' (see also: Rouse, 1981; Dasgupta, 1991).

This concept of performance ultimately supports the evaluation of worksystems. For example, by relating task quality to resource costs we are able to distinguish between two different designs of cognitive tool which, whilst enabling the same goals to be achieved, demand different levels of the user's resource costs. The different performances of the two worksystems which embody the tools would therefore be discriminated. Similarly, think about the implications of this concept of performance for concern with user error: it is not enough for user behaviours simply to be error-free; although eliminating errorful behaviours may contribute to the best performance possible, that performance may still be less than desired. On the other hand, although user behaviours may be errorful, a worksystem may still achieve a desirable performance. Optimal human behaviour uses a minimum of resource costs in achieving goals. However, optimality can only be determined categorically against worksystem performance, and the best performance of a worksystem may still be at variance with the performance desired of it.

This concept of performance allows us to recognise an *economics of performance*. Within this economy, structural and behavioural costs may be traded-off both within

and between the agents of the worksystem, and those costs may be traded off also with task quality. Users may invest structural costs in training the cognitive structures needed to perform a specific task, with a consequent reduction in the behavioural costs of performing that task. Users may expend additional behavioural costs in their work to compensate for the reduced structural costs invested in the under-development of their cognitive tools.

The economics of worksystem performance are illustrated by Sperandio's observation of air traffic controllers at Orly control tower (Sperandio 1978). Sperandio observed that as the amount of traffic increased, the controllers would switch control strategies in response to increasing workload. Rather than treating each aircraft separately, the controllers would treat a number of following aircraft as a chain on a common route. This strategy would ensure that safety for each aircraft was still maintained, but sacrificed consideration of time keeping, fuel usage, and other expedition goals. This observation of the controllers' activity can be understood as the controller modifying their (generic) behaviours in response to the state of the domain as traffic increases. In effect, the controller's are trading-off their resource costs, that is, limiting their workload, against less critical aspects of task quality. The global effect of modifying their behaviour is a qualitative change in worksystem performance. Recent work in modelling air traffic management (Lemoine and Hoc, 1996) aims to dynamically re-distribute cognitive work between controllers and tools in order to stabilise task quality and controller resource costs, and therefore to stabilise worksystem performance.

2.7 Cognitive design problems

Engineering disciplines apply validated models and principles to prescribe solutions to problems of design. How then should we conceive of the design problems which Cognitive Engineering is expected to solve? It is commonplace for cognitive design to be described as a 'problem solving activity', but such descriptions invariably fail to say what might be the nature and form of the problem being solved. Where such reference is made, it is usually in domain specific terms, and a remarkable variety of cognitive design problems is currently presented, ranging from the design of teaching software for schools to the design of remote surgery. A recent exception can be found in Newman and Lamming (1995). Yet the ability to acquire knowledge which is valid from one problem to the next requires an ability to abstract what is general from those two problems. We presume that instances of cognitive design problems each embody some general form of design problem and further, that they are capable of explicit formulation. The following proposes that general form.

Cognitive work can be conceptualised in terms of a worksystem and a domain and their respective concepts. In performing work, the worksystem achieves goals by transformations in the domain and it also incurs resources which have their cost (Figure 1). The aim of design is therefore 'to specify worksystems which achieve a desired level of performance in given domains'.

More formally, we can express the general design problem of Cognitive Engineering as follows:

> Specify then implement the cognitive structures and behaviours of a worksystem {W} which performs work in a given domain (D) to a desired

level of performance (P) in terms of task quality (ΣQ) and cognitive user costs (ΣK_U).

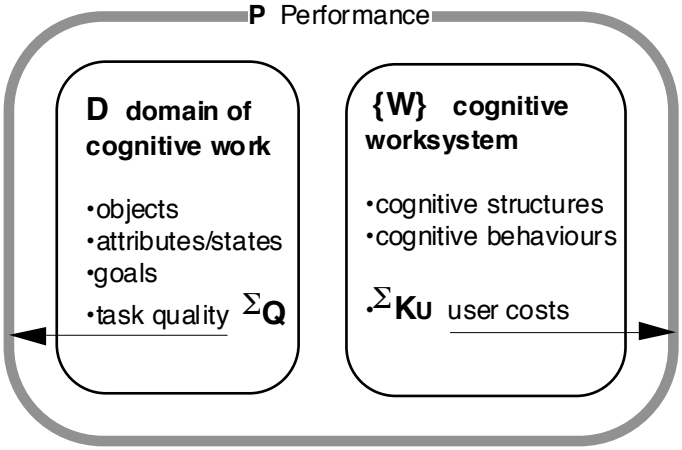

Figure 1. Worksystem and a domain

An example of such a cognitive design problem formulated in these terms might refer to: the requirement for specifying then implementing the representations and processes as the knowledge of an air traffic management worksystem which is required to manage air traffic of a given density with a specified level of safety and expedition and within an acceptable level of costs to the controllers. This problem expression would of necessity need to be supported by related models of the air traffic management worksystem and domain (see Dowell, in prep).

By its reference to design practice as 'specify then implement', this expression of the general cognitive design problem is equivalent to the design problems of other engineering disciplines; it contrasts with the trial and error practices of craft design. However, the relationship between the general cognitive design problem and the design problems addressed by other engineering disciplines associated with the design of cognitive tools, such as Software Engineering and 'Hardware Engineering', is not explicitly specified. Nevertheless, it is implied that those other engineering disciplines address the design of the internal behaviours and structures of cognitive tools embedded in the worksystem, with concern for the resource costs of those tools.

3. Prospect of Cognitive Engineering principles

The deficiencies of current cognitive design practices have prompted our investigation of Cognitive Engineering as an alternative form of discipline. Our analysis has focused on the disciplinary matrix of Cognitive Engineering consisting of a conception, values, design principles and exemplars. The analysis assumes that Cognitive Engineering can make good 'the deficiencies'. First, the integration of cognitive design in systems development would be improved because Cognitive Engineering principles would enable the formulation of cognitive design problems and the early prescription of design solutions. Second, the efficacy of cognitive design would be improved because Cognitive Engineering principles would provide

the guarantee so lacking in cognitive design which relies on experiential knowledge. Third, the efficiency of cognitive design would be improved through design exemplars related to principles supporting the re-use of knowledge. Fourth, the progress of cognitive design as a discipline would be improved through the cumulation of knowledge in the form of conception, design principles and exemplars.

However, we observe that these elements of the disciplinary matrix required by Cognitive Engineering remain to be established. And since not all are likely to be established at the same time, the question arises as to which might be constructed first. A conception for Cognitive Engineering is a pre-requisite for formulating engineering principles. It supplies the concepts and their relations which express the general problem of cognitive design and which would be embodied in Cognitive Engineering principles.

To this end, we have proposed a conception for Cognitive Engineering in this paper, one which we contend is appropriate for supporting the formulation of Cognitive Engineering principles. The conception for Cognitive Engineering is a broad view of the Cognitive Engineering general design problem. Instances of the general design problem may include the development of a worksystem, or the utilisation of a worksystem within an organisation. Developing worksystems which are effective, and maintaining the effectiveness of worksystems within a changing organisational environment, are both expressed within the problem.

To conclude, it might be claimed that the craft nature of current cognitive design practices are dictated by the nature of the problem they address. In other words, the indeterminism and complexity of the problem of designing cognitive systems (the softness of the problem) might be claimed to preclude the application of prescriptive knowledge. We believe this claim fails to appreciate that the current absence of prescriptive design principles may rather be symptomatic of the early stage of the discipline development. The softness of the problem needs to be independently established. Cognitive design problems are, to some extent, hard: human behaviour in cognitive work is clearly to some useful degree deterministic, and sufficiently so for the design, to some useful degree, of interactive worksystems.

The extent to which Cognitive Engineering principles might be realiseable in practice remains to be seen. It is not supposed that the development of effective systems will never require craft skills in some form, and engineering principles are not incompatible with craft knowledge. Yet the potential of Cognitive Engineering principles for the effectiveness of the discipline demands serious consideration. The conception presented in this paper is intended to contribute towards the process of formulating such principles.

Acknowledgement. We acknowledge the critical contributions to this work of our colleagues, past and present, at University College London. John Dowell and John Long hold a research grant in Cognitive Engineering from the Economic and Social Research Council.

References

Ashby W. R., (1956). *An introduction to cybernetics*. Methuen: London.

Barnard P. and Harrison M., (1989). Integrating cognitive and system models in human computer interaction. In: Sutcliffe A. and Macaulay L. (ed.s). *People and*

Computers V. Proceedings of the Fifth Conference of the BCS HCI SIG, Nottingham 5-8 September 1989. Cambridge University Press, Cambridge.
Beltracchi L., (1987). A direct manipulation interface for water-based rankine cycle heat engines, IEEE transactions on systems, man and cybernetics, SMC-17, 478-487.
Card, S. K., Moran, T., Newell, A., (1983). *The Psychology of Human Computer Interaction*. Erlbaum: New Jersey.
Carroll J.M., and Campbell R. L., 1986, Softening up Hard Science: Reply to Newell and Card. *Human Computer Interaction*, 2, 227-249.
Checkland P., (1981). *Systems thinking, systems practice*. John Wiley and Sons: Chichester.
Dasgupta, S., (1991). *Design theory and computer science*. Cambridge University Press: Cambridge.
Dowell J. and Long J.B., (1989). Towards a conception for an engineering discipline of human factors. In*Ergonomics*, 32, 11, pp 1513-1535.
Dowell, J., (in prep) The design problem of Air Traffic Management as an examplar for Cognitive Engineering.
Hoare C.A.R. , 1981. Professionalism. *Computer Bulletin*, September 1981.
Hoc J.M., (1990). Planning and understanding: an introdution. In Falzon P. (ed.). *Cognitive Ergonomics: Understanding learning and designing human computer interaction.* Academic Press: London.
Hollnagel E. and Woods D.D., (1983). Cognitive systems engineering: new wine in new bottles. *International Journal of Man-Machine Studies*, 18, pp 583-600.
Hollnagel E., (1991). The phenotype of erroneous actions: implications for HCI design. In Alty J. and Weir G. (ed.s), *Human-computer interaction and complex systems*. Academic Press: London.
Hutchins, E. (1994) Cognition in the wild Mass: MIT press.
Johnson J., Roberts T., Verplank W., Irby C., Beard M. and Mackey K., (1989). The Xerox Star: a retrospective. *IEEE Computer*, Sept, 1989, pp 11-29.
Just M.A. and Carpenter P.A., 1992 A capacity theory of comprehension: individual differences in working memory, *Psychological Review*, 99, 1, 122-149.
Kuhn T.S., (1970). *The structure of scientific revolutions*. 2nd edition. University of Chicago press: Chicago.
Landes D.S., (1969). *The unbound prometheus*. Cambridge University Press: Cambridge.
Layton E., (1974). Technology as knowledge. *Technology and Culture*, 15, pp 31-41.
Lemoine M.P. and Hoc J.M., (1996) Multi-level human machine cooperation in air traffic control: an experimental evaluation. In Canas J., Green T.R.G. and Warren C.P (ed.s)*Proceedings of ECCE-8. Eighth European Conference on Cognitive Ergonomics*. Granada, 8-12 Sept, 1996.
Lenorovitz, D.R. and Phillips, M.D., (1987). Human factors requirements engineering for air traffic control systems. In Salvendy, G. (ed.) *Handbook of Human Factors*. Wiley, London. 1987.
Long J.B. and Dowell J., (1989). Conceptions of the Discipline of HCI: Craft, Applied Science, and Engineering . Published in: Sutcliffe A. and Macaulay L. (ed.s). *People and Computers V*. Cambridge University Press, Cambridge.
Long J.B. and Dowell J., (1996). Cognitive Engineering human computer interactions *The Psychologist*., Vol 9, pp 313 - 317.
McShane J., Dockrell J. and Wells A., (1992). Psychology and cognitive science. In *The Psychologist*:, 5, pp 252-255.
MumfordE. 1995. *Effective requirements analysis and systems design: the ETHICS method*. Macmillan

Neisser U., 1987. From direct perception to conceptual structure. In U. Neisser, *Concepts and conceptual development: ecological and intellectual factors in categorisation*, CUP.
Newman W. and Lamming M., 1995,*Interactive System Design*. Addision-Wesley.
Norman D.A., (1986). Cognitive engineering. In Norman D.A. and Draper S.W., (ed.s)*User Centred System Design*. Erlbaum: Hillsdale, NJ.
Phillips M.D. and Melville B.E., (1988). Analyzing controller tasks to define air traffic control system automation requirements. In *Proceedings of the conference on human error avoidance techniques*, Society of Automotive Engineers. Warrendale: Penn.. pp 37-44.
Phillips M.D. and Tischer K., (1984). Operations concept formulation for next generation air traffic control systems. In Shackel B. (ed.), *Interact '84*, Proceedings of the first IFIP conference on Human-Computer Interaction. Elsevier Science B.V.: Amsterdam. pp 895-900.
Rasmussen J. and Vicente K., (1990). Ecological interfaces: a technological imperative in high tech systems? *International Journal of Human Computer Interaction,* 2 (2) pp 93-111.
Rasmussen J., Pejtersen A., and Goodstein L., (1994) *Cognitive Systems Engineering*. New York: John Wiley and Sons.
Rouse W.B., (1980). *Systems engineering models of human machine interaction*. Elsevier: North Holland.
Shaw M., (1990) Prospects for an engineering discipline of software. IEEE Software, November 1990.
Simon H.A., (1969). *The sciences of the artificial*. MIT Press: Cambridge Mass..
Sperandio, J.C., (1978). The regulation of working methods as a function of workload among air traffic controllers. *Ergonomics*, 21, 3, pp 195-202.
Vicente K., (1990). A few implications of an ecological approach to human factors. *Human Factors Society Bulletin*, 33, 11, 1 - 4.
Vincenti W.G. (1993) *What engineers know and how they know it*. John Hopkins University Press: Baltimore.
Winograd T. and Flores F., (1986). *Understanding computers and cognition*. Addison Wesley: Mass..
Woods D.D. and Roth E.M., (1988). Cognitive systems engineering. In Helander M. (ed.) *Handbook of Human Computer Interaction*. Elsevier: North-Holland.
Woods D.D., (1994). Observations from studying cognitive systems in context. In Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society (Keynote address).

affordance - the potential change in state of a domain object

agent - a sub-system whose behaviours are those of a single user or computer

attribute states - the specific 'values' of attributes

attributes - the 'properties' of domain objects

behaviours - activities of agents and tools which we recognise as work being performed.

boundary - the extent of all user and device behaviours whose intention is to achieve the same goals in a given domain

constants - attributes with states that are fixed

domain - a task world whose possibilities and constraints are organised around specific goals

economics of performance- the trade-off between resource costs and task quality that may occur both when worksystems perform work, and in design

external behaviour - worksystem behaviour which is expressed at the worksystem boundary and which directly transforms domain objects

goals - specific desirable states of attributes to be achieved

hierarchy of complexity - the levels of agent behaviours and structures and the relations between them

interaction - the coupling of two agents

internal behaviours - worksystem behaviours which are not expressed at the worksystem bounday

objects - the elemental constituents of domains, identified by their attributes

performance - the relationship between the task quality achieved in performing work and the resource costs incurred

processes- structures which, when activated, transform (process) information

quality - the extent to which goals are achieved through state changes in the attributes of domain objects

representations - structures which hold information

resource costs - the costs to agents of establishing structures and expressing behaviours

structures - the components of agents and tools which give rise to their behaviour

states - values of the attributes of objects in a domain

sub-systems - functional groupings of worksystem behaviours which may be contributed by more than one agent.

task quality - the difference between the goals and the actual state changes achieved through work

user interface - the combination of structures of both user and computer supporting their interacting behaviours

variables - attributes with states that change

work - intended and realised changes in the attribute states of objects

worksystem - the system of agents interacting with each other to perform work by intentionally changing the states of domain objects