# COGNITIVE ENGINEERING AND HUMAN-COMPUTER INTERACTIONS

*Dr. Andrew Ajegba Ekoja*

**Abstract**

4

**The increase in cognitive work brought about by the use of computers presents ' new opportunities for Cognitive Psychology and Engineering. In this article, it is considered how cognitive psychology might relate to "cognitive engineering design', to support the development of computing technology. The first, and established, one-stream view sees an applied cognitive psychology making direct use of theories developed by basic science, and more recent, two-stream view sees a discipline of cognitive engineering , separate from that of cognitive psychology, equivalent to software engineering. The paper discusses the implications of the two-stream view.**

## Engineering Psychology

Engineering and psychology may seem an unusual marriage. After all, engineering is about the design of technological systems, and psychology about understanding the human mind and behaviour. But partners they are expected to be, thanks to the 'shot-gun' wedding regularly prompted by technological innovation. The typical scenario is as follows. A new technology is born, for example, the sensing and electro-mechanical capability developed during World War Two. The new technology is designed and implemented by engineering into technical systems, for example, radar and sonar, etc. Yet because of new problems associated with their users, these systems often fail to produce the expected gains in effectiveness. For example, they produced inefficient radar and sonar monitoring, which resulted from boredom and distraction, Psychology was then brought in to help solve these operational problems, for example, the 'vigilance' problem, of missed sonar signals late in the submariner's watch, was shown to be affected by operator selection and training; display design; working procedures, etc. Such 'shot-gun' weddings seem to be a consistent feature of the relationship between technology, engineering and psychology. Cognitive engineering is the most recent attempt to regularise 'shot-gun' aspects of the marriage between engineering and psychology, and one arising from the developments of interactive computing.

## Psychology and IT

There can be little doubt that we are now experiencing an information revolution, a revolution driven by advances in computer technology. Such advances include personal computers, electronic networks, multi-media displays, integrated communications, and data processing, etc. Software engineering has designed and implemented this computer technology into technical systems used in hospitals, businesses, emergency services, government administration, etc. However, such systems frequently exhibit operation problems associated with the users, for instance the unexploited functionality of video recorders. So if once the main design issue was the computer and 'getting it to work at all, the main issue might now be said to be human-computer interaction (HCI) and getting users interacting with computers to perform effective work. Again psychology, under the guise of cognitive psychology, has been called in to help on offer, and indeed, so difficult as to bring into question what is, or should be, the relationship between interactive information technology (IT), software engineering and cognitive psychology.

The established view of this relationship, the so-called one-stream view, distinguishes a pure and an applied science, wherein the theory provided by pure science trickles (or is pulled) down into applied science (hence, one stream). In this way, applied cognitive psychology is assumed to use scientific theory to improve the effectiveness of cognitive work and cognitive tools that support that work. This view assumes a direct relationship between cognitive psychology, applied cognitive psychology and IT, although it is silent about the relationship with software engineering. The view usually remains





unchallenged, in spite of how little evidence there is that the theories provided by cognitive psychology have directly contributed to the design of IT.

However, there is a different view of the relationship, the so-called two-stream view, which acknowledges the possibility of cognitive design as a discipline in its own right: a discipline of cognitive engineering more equivalent to software engineering, and separate from cognitive psychology. This view makes no assumptions about the necessary uptake of theory provided by cognitive psychology (hence the two-stream view), although the possibility of such uptake, if appropriate, by a theory of cognitive engineering is not excluded.

There are critical implications of these alternative one and two-stream views of the relationship between interactive IT, software engineering and cognitive psychology. For example, an implication for research concerns whether the theory needed for the design of effective information systems can be obtained by direct application of cognitive psychology, or whether the theory needs to be constructed anew, more directly from the design problems being addressed, such a problem would be to design an interactive management system for Nigerian air traffic to support a 30 per cent increase in traffic with no reduction in safety.

The following sections discuss and contrast the two views with reference to two research initiatives in the area, the Joint Council's Initiative in Cognitive Science and Human-Computer Interaction, and the ESRC programme in Cognitive Engineering. These initiatives provide a timely opportunity to illustrate the two views and constitute important research opportunities to progress the field. The implications of viewing cognitive engineering as a discipline of design are then distinguished and illustrated with reference to the ESRC programme.

**Applied Cognitive Psychology**

Cognitive psychology is a science. Its aim is to understand behaviours such as perception, reasoning, communication, action, etc. in terms of their relationship to knowledge. It provides this understanding through theories which typically take the form of models of mental representations and processes. Such models are used to explain and to predict empirical findings. Psycholinguistics is one area of cognitive psychology and one can reasonably assume that, because users communicate with computers through language, their dialogue exhibits general psycholinguistic phenomena, for example, the dialogue between the air traffic controller and the pilot. Yet it is here that there occurs a disjunction in the argument: simply, it is implicit in the one-stream, applied science approach that theories of psycholinguistics can directly support the design of human-computer dialogue. More generally, it is implicit in the applied science approach that applying cognitive psychology theory will contribute to users interacting with computers to perform more effective work, for example, managing 30 percent more Nigerian air traffic with no reduction in safety.

This approach informed the Joint Council's Initiative (JCI) in the United Kingdom which, although concerned more generally with cognitive science, include cognitive psychology (along with linguistics, artificial intelligence, etc). The aims of the JCI, an example of the one-stream approach, were to enhance our understanding of the general computational principles underlying natural and artificial forms of intelligence, and their application in the design of systems involving human-computer interaction.

The research programmes was divided into four areas: systems design; principles of interaction; compute rational learning environments; and computational modeling of cognition. It is generally agreed that the initiative was successful in supporting important advances in cognitive science; it would appear to have been less successful however, in promoting the application of cognitive science to HCI (an outcome unfavourable to the one-stream approach). Although many reasons could be suggested for this outcome, we will restrict ourselves here to those reasons which concern the applied science approach of cognitive psychology. The reasons can be found by consideration of the discipline of cognitive psychology, and are as follows.

We might characterize a discipline as knowledge supporting practices which address a general problem having a particular scope (following Long and Dowell, 1989). Cognitive psychology has models



 laws, theories, etc) as its knowledge, explanation and prediction as its practices, understanding as its general problem, and human minds and behaviour as its particular scope.

It naturally follows that cognitive psychology can provide an understanding of users interacting with computers, for examples, the dialogue between the air traffic controller and the pilot. Yet it can be reasonably argued that what is required, to realize the potential of IT, is the design of technical systems (cfi, the role of software engineering), for example, the specification and evaluation of the air traffic management system itself. It is unclear how the knowledge and practices of cognitive psychology, concerned with the general problem of understanding, are able to address the general problem of design of the air traffic management system. If we accept that these are, in fact, quite different (although possibly related) problems, we must also accept that they are unlikely to do so. For example, a psycholinguistic model of air traffic controller/pilot dialogue would not itself reference the effectiveness of traffic management (and so would have no explicit relationship with the desired increase of 30 percent). In contrast, the second approach to HCI, that of cognitive engineering as a design discipline, addresses design as its general problem.

**Cognitive Engineering**

Cognitive engineering as a discipline can also be characterized as knowledge supporting practices which address a general problem having a particular scope. The general problem of cognitive engineering, however, is design, the particular scope of this problem is users interacting with computers to perform work effectively.

A symmetry of software engineering and cognitive engineering is clear. Software engineering designs the computers as they interact with users, for example, the electronic flight strips and radar of air traffic controllers. Cognitive engineering designs the users as they interact with computers, for examples, the design (and training) of the managing behaviour, such as planning, etc, to control the air traffic. Together they design users interacting with computers to perform effective work, that is an air traffic management system which solves the design problem of increasing air traffic by 30 percent with no reduction in safety.

This one-stream approach would appear to have informed, at least in part, the recent ESCR'S research programme in cognitive engineering. The main objectives of the programme is to bring people, as individuals and as groups, into the heart of research on design problems (ESRC, 1995:1). Its specific objectives include: to acquire new cognitive engineering knowledge; to feed back the results (to cognitive and social sciences); to encourage application of new cognitive engineering knowledge (by systems developers); and to disseminate new cognitive engineering knowledge (to users) (ESRC, 1995:1).

In the remainder of this article, general issues are considered deriving from the view of cognitive engineering as a discipline of design in its own right (Proponents of the one-stream view might like to undertake a comparable exercise.)

**Design Discipline**

The issues are grouped around the characterization of a discipline provided earlier knowledge supporting practices which address a general problem having a particular scope. This characterization ensures coherence and completeness of address. The issues are posed as questions to which some response are offered (see also Long and Dowell, 1989; Dowell and Long, 1989; and Dowell, 1993). References to alternative responses are also provided. To make access and referencing easier, the questions are numbered.

**Discipline**

Cognitive engineering as a design discipline raises a number of issues at the level of the discipline itself. These issues are as follows:

i. If cognitive engineering is engineering and separate from cognitive psychology, what general engineering notion should cognitive engineering embody?



Notions of cognitive engineering vary from temporary fixing (Norman, 1986) to providing exact solutions to design problems (Dowell and Long, 1989). We have suggested the general notion of 'engineering for effectiveness, wherein the aim is to provide systems which at least perform the work as well as we require and at the cost we find acceptable. Although the ESRC programme does not make an explicit commitment to a particular form of cognitive engineering, it is clear that it believes systems must be devised to help people and computer work effectively (ESRC, 1995:3. The commitment to effectiveness and to a particular notion of cognitive engineering is significant, since it must surely be constrained by the notion of engineering to which software engineering subscribes. For example, cognitive and software together would be expected to design a more effective management system for UK air traffic, and not just the one or the other.

ii. If cognitive engineering is committed to design for effectiveness, what view of design should cognitive engineering espouse?

Design disciplines vary in their view of design, from technocratic art through craft (Winograd & Flores, 1986) to explicit specification of design solutions (Dowell and Long, 1989) We have proposed a formal view requiring the technical expression of design problems and solutions. Interestingly, the ESRC programme describes cognitive engineering as model(ling) the design problems associated with interactive cognitive systems (ESRC, 1995:2). Whilst we may understand here a broad view of modeling, it is nevertheless clear that the programme seeks to promote a more formal discipline. This decision is important, since it has often been argued that the knowledge currently used to design human-computer interactions is principally the experience and insight of the practitioner. Hence, it might be suggested that models of design problems could in the short term be derived from practitioner knowledge. For example, UK air traffic controllers could contribute to a formation of the UK air traffic management design problem of increasing traffic by 30 percent.

iii Given a formal view of cognitive engineering, what genera forms of cognition should cognitive engineering include?

Cognition varies from individuals to groups (distributed or social cognition). We have suggested that cognition should be knowledge-intensive, and apply to individual users, networked users, and computers. Similarly, the ESRC initiative treats both users and computers as interactive cognitive systems. It intentionally does not confine itself to either individual or collective cognitive systems, but rather stresses the requirement for multi-level approach to include the micro and macro-features of interactive cognitive working. This intention is significant because some of the most demanding challenges facing cognitive engineering have arisen from the penetration of computing technology into shared or organizational structures and communications. The safety culture of UK air traffic management teams, which sustains current high standards of safety, provides a good illustration.

The next set of issues concern the design problem addressed by cognitive engineering.

**Design problems.**

iv Given its independence, as a discipline, from cognitive psychology, how should cognitive engineering characterize its design problems?

Design problems vary in their generality, in their technical expression, and in their scope. It has been suggested that design problems should be formally and technically expressed. We have proposed a technical conception intended to support a more coherent and complete expression of the general design problem (Dowell and Long, 1989). The ESRC programme, by contrast, tends to refer broadly to design problems, without prejudice to the importance of either particular or general design problems, or to their explicit formulation. This reference is significant because the processes of generalization, validation and application (see later) depend on the generality and explicitness of the design problems to which they relate. For example, the general UK air traffic management design problem would be made up of more specific individual UK airport and sector design problems as they relate to the effective management of the traffic.

The next set of issues concern the particular scope of the general problem of cognitive engineering.

v Given its general design problem what should be the scope of the general design problem of cognitive engineering? *

Scope can vary from consideration of only the computing device (Sommerville, 1992) to the system, organization and environment in which it operates (Eason, 1988). We have proposed an informal expression of the scope to be users interacting with computers to perform effective work (Dowell and Long, 1989) - the expression used here throughout. The ERSC programme appears to take much the same view of the scope of cognitive engineering, in particular, it is strong on the need to describe user interactions in the connect of the user's work and their organizational situation. This commitment is significant since it appears to recognize the limited success of earlier attempts to develop universal models and methods, and the need for more domain-specific theory to support cognitive design, for example, a cognitive design theory able to specify air traffic controllers interacting with electronic flight strips and radar to manage 30 percent more UK air traffic.

The next set of issues concern the cognitive engineering knowledge intended to support the solution of the general design problem.

**Knowledge**

vi Given the general design problem and its scope, what form(s) should cognitive engineering knowledge take and what guarantees should it offer?

Knowledge could vary in form: heuristics (Whiteside, & Holtzbatt, 1988); guidelines (Smith and Mosier, 1986); laws (Card, Moran and Newell, 1983); methods (Lim and Long, 1994); standards; etc. Knowledge could also vary in the guarantee with which, when applied, it offers a solution to the design problem, from best guess (with many subsequent iterations) Shneiderman, 1992) to full 'guarantee' (Dowell and Long, 1989). We have proposed that cognitive engineering knowledge takes the form of substantive (addressing the what) and methodological (addressing the how) design principles. The design principles would be conceptualized; operationalized; tested; and generalized. They would, thus, support the derivation of a general solution to the general design problem of cognitive engineering with guarantee, for example, a UK air traffic control system managing 30 percent more air traffic with no reduction in safety.

Similarly, the ESRC programme refers to engineering principles, and to a more systematic and principled approach that enables verification, generalization and validation of the knowledge employed. This reference is important, as it seems to recognize the need for direct, rather than indirect, support for cognitive design, and more effective support.

vii Given the two-stream view, what relations should hold between cognitive engineering knowledge and other types of knowledge, especially that of cognitive psychology?

A complete range of relations might in principle exist between cognitive engineering knowledge and other types of knowledge from implicit relationships (Dowell and Long, 1989) to explicit co-identity (Barnard, 1991). We have proposed that any knowledge on be eligible for use in the generation of cognitive engineering knowledge, but that the later must support solution of the general design problem. The relation would also hold for cognitive psychology knowledge. If it were to prove useful in the construction of design principles, then it would be recruited; otherwise not (hence, the two-stream view). Further, even if recruited it would require transformation, since the general problem of understanding (for example, current, air traffic controller behaviour) is not the same as the general problem of design (for example, specifying new air traffic control behaviour to manage 30 percent more air traffic)(see earlier). According to the ERSC programme:

> **The link between cognitive science (including cognitive psychology is complex. However, it is generally agreed that results from cognitive science have a significant role to play in advancing the state of cognitive engineering. In particular, we may view cognitive science as a supplier of architecture for human and computers,; cognitive engineering then deploys and independently. develops these architecture to express and solve design problems (ESRC, 1995:3).**

Although the reference to independent development is significant, more than architectures for human and computers (for example, a conception and principles for cognitive design) will be required to express and solve design problems. For example, an architecture for air traffic controllers would have to be tightly coupled to management effectiveness to be able cognitively do design and engineer 30 percent more air traffic. In addition, the programme postulates a relation with cognitive engineering knowledge to feed back the results of the research so that it contributes to the fundamental understanding of the cognitive and social sciences (ESRC, 1995 :1). This would appear to us to be a noval requirement.

The next and last set of issues concern the practices of cognitive engineering.

**Practice**

viii Given the knowledge of cognitive engineering, what types of practice should cognitive engineering espouse?

Practices are related intimately to the type of general design problem, and the knowledge recruited to its solution. However, such practices vary widely for example, trial and error (Whiteside et al., 1988), 'specify and implement' (Long and Dowell, 1989), etc. we have proposed that design practice be of the 'pecify-then-implemenf (Dowell and Long, 1989) kind. That is, design principles support the specification of a design solution in advance of its implementation (and with guarantee - see earlier). The practice derives from, and is justified by, the known relation between general design problem and general design solution (as design principles). For example, the increase in UK air traffic by 30 per cent and the design of a control system to manage that system with no reduction in safety. The ESRC programme cites concern for evaluation techniques, advocating in addition a two-step practice of first investigating the cognitive and social constraints on the use of existing technological devices and second incorporating knowledge of these constraints into the design of new devices (ESRC, 1995:2).

So, we have now covered each of the areas where questions are raised by the idea of cognitive engineering as a design discipline. To recap, those questions relate to the discipline; its design problems; its problem scope its knowledge and it practice.

**Conclusion**

An attempt has been made here to characterize cognitive engineering and its relationship to IT and software engineering. The major points can be briefly summarized. First, cognitive engineering, as a design discipline, the two-stream view, is a novel approach to getting users interacting with computers to perform effective work. It can be distinguished from the approach to cognitive psychology as an applied science, the one-stream view. Second, the important novelty of cognitive engineering is two-fold. It has its own knowledge and practices. In addition, because its general problem is design, its solutions should be able to offer some guarantees. The emergent discipline, however, has yet to show its paces. For many years, it has been an idea looking for development (Norman, 1986; Rasmussen, 1986; Hollnagel and Woods, 1983). The ERSC's research programme in cognitive engineering presents a great opportunity to contribute to that development.

**References**

Barnard, P (1991) Bridging Between Basic Theories and the Artifacts of Human-Computer Interaction, in J. Carrol (ed); *Designing Interaction: Psychology at the Human Computer Interface* Cambridge, MA: MIT.

Card, S. K., Moran, T. P. and Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Dowell, J. (1993) *Cognitive Engineering and the Rationalization of the Flight Strip,* Ph.D Thesis.

Dowell J. and Long, J. B. (1989). Towards a Conception for an Engineering Discipline of Human Factors *Ergonomics 32(11),* 11513 - 1535.

Eason, K. D. (198) *Information Technology and Organizational Change.* London: Taylor & Francis.

ESRC (1995) *Cognitive Engineering Research Programme: Programme Specification Economic and Social Research Council.* Swindon, UK.

Hollnagel, E. and Woods, SD. D. (1993) Cognitive Systems Engineering: New Wine in new Bottles *International Journal of Man-Machine Studies* 18, 583 - 600.

Lim, K. Y and Long, J. B. (1994) *The MUSE Method For Usability Engineering* Cambridge, UK. Cambridge University Press.

Long, J. B and Dowell, J. (1989). Conceptions for the Discipline of HCI Craft Applied Science, and Engineering. In A Sutclife and Macaulay (eds): *People and Computers* V. Proc. Fifth Conference of the BCS HCL SIG, Nottingham, 5-8 September 1989. Cambridge: Cambridge University Press.

Norman, D. (1986) Cognitive Engineering in D. A. Norman and S. W. Draper: *User Centered System Design.* Hillsdale, NJ: Erlbaum.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction: An Approach in Cognitive Engineering.* New York: North-Holland.

Shnederman, B. (1992) Designing the User Interface Strategies for Effective Human Computer Interaction (end ed). *Reading, MA:* Addison Wesley.

Smith, S. L. and Mosier J. N. (1986) *Guidelines for designing User Interface Software* ESD-TR86-278. Bedford M. A: MITRE Corporation.

Sommerville, I. (1992) *Software Engineering* (4$^{th}$ ed). Workingham, UK: Addison-Wesley.

Whiteside, J., Bennett, J. and Holtzbatt, . (1988) Usability Engineering: Our Experience and Evolution. In M. Helander (Ed): *Handbook of Human Computer Interaction.* Amsterdam. North Holland.