Socio-cognitive Engineering

Mike Sharples

Mike Sharples. Socio-cognitive Engineering. C. Ghaoui. Encyclopedia of Human-Computer Interaction, IGI Global, pp.542-547, 2006, Encyclopedia of Human-Computer Interaction. hal-00190312

Socio-cognitive Engineering

Mike Sharples
Educational Technology Research Group
University of Birmingham
m.sharples@bham.ac.uk

ABSTRACT

Socio-cognitive engineering is a framework for the human-centred design of technology-based systems to enhance human knowledge working, decision making, collaboration and learning. Like user-centred design, it draws on the knowledge of potential users and involves them in the design process. But it extends beyond individual users to analyse the activity systems of people and their interaction with technology, including their social interactions, styles and strategies of working, and language and patterns of communication, to form a composite picture of human knowledge and activity that can inform system design. The framework consists of two main parts: a phase of *activity analysis* to interpret how people work and interact with their current tools and technologies, and a phase of *systems design* to build and implement new interactive technology. Socio-cognitive engineering has been refined and tested through a series of projects to develop computer systems for supporting learning and knowledge working.

INTRODUCTION

Socio-cognitive engineering is a framework for the systematic design of socio-technical systems (people and their interaction with technology) based on study and analysis of how people think, learn, perceive, work and interact. The framework has been applied to the design of a broad range of human centred technologies, including a Writer's Assistant (Sharples, Goodlet, & Pemberton, 1992) a training system for neuroradiologists (Sharples et al., 2000), and a mobile learning device for children (Sharples, Corlett, & Westmancott, 2002). It has been adopted by the European MOBIlearn project (www.mobilearn.org) to develop mobile technology for learning. It has also been taught to undergraduate and postgraduate students, to guide their interactive systems projects. An overview of the framework can be found at (Sharples, Jeffery et al., 2002).

BACKGROUND

The approach of socio-cognitive engineering is similar to user-centred design (Norman & Draper, 1986) in that it builds on studies of potential users of the technology and involves them in the design process. But users are not always reliable informants. They may idealize their methods, describing the ways in which they would like to or have been told to work, rather than their actual practices. Although users may be able to describe their own styles and strategies of working, they may not be aware of how other people can perform a task

differently and possibly more effectively. Surveys of user preferences can result in new technology that is simply an accumulation of features, rather than an integrated system.

Thus, socio-cognitive engineering is critical of the reliability of user reports. It extends beyond individual users to form a composite picture of the human knowledge and activity including cognitive processes and social interactions, styles and strategies of working, and language and patterns of communication. The term 'actor' rather is used rather than 'user', to indicate that the design may involve people who are stakeholders in the new technology but are not direct users of it.

The framework extends previous work in soft systems (Checkland & Scholes, 1990) socio-technical and cooperative design (Greenbaum & Kyng, 1991; Mumford, 1995; Sachs, 1995) and the application of ethnography to system design (see (Rogers & Bellotti, 1997) for a review). It incorporates existing methods of knowledge engineering, task analysis and object-oriented design, but integrates them into a coherent methodology that places equal emphasis on software, task, knowledge and organizational engineering.

The framework also clearly distinguishes studying everyday activity using existing technology from studying how the activity changes with proposed technology. It emphasises the dialectic between people and artefacts: using artefacts changes people's activities, and this in turn leads to new needs and opportunities for design.
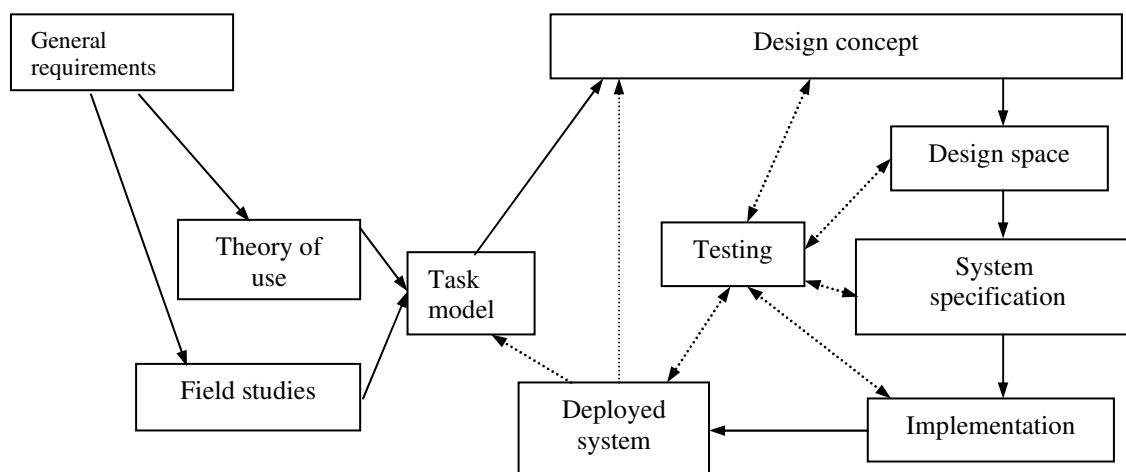

Figure 1. Overview of the flow and main products of the design process.

FRAMEWORK

Figure 1 gives a picture of the flow and main products of the design process. It is in two main parts: a phase of *activity analysis* to interpret how people work and interact with their current tools and technologies, and a phase of *systems design* to build and implement new interactive technology. The bridge between the two is the relationship between the Task Model and the Design Concept. Each phase comprises *stages* of analysis and design, which are implemented through specific *methods*. The framework does not prescribe which methods to use: the choice depends on the type and scale of project.

It is important to note that the process is not a simple sequence, but involves a dialogue between the stages. Earlier decisions and outcomes may need to be revised to take account of later findings. When the system is deployed it will enable and support new activities, requiring another cycle of analysis, revision of the Task Model and further opportunities for design.

The elements of socio-cognitive engineering are shown below.

Project. The diagram shows the process of design, implementation and deployment for a single project.

Actors. Different types of people may be involved in or affected by the design and deployment, including (depending on the scale of the project) design, marketing and technical support teams, direct users of the system, and other people affected by it (such as administrative staff).

Roles. The actors take on roles (such as 'team leader'), that may change during the project.

Stage. Each box represents one stage of the project.

Methods. Each stage can be carried out by one or more methods of analysis and design, which need to be specified before starting the stage.

Tools. Each method has associated tools (for activity analysis, software specification, systems design, and evaluation) to carry out the method.

Outcomes. Each stage has outcomes which must be documented, and these are used to inform and validate the system design.

Measures. Each design decision must be validated, by reference to outcomes from one of the stages.

The general sequence for socio-cognitive engineering is as follows:

1. Form a project team.
2. Produce general requirements for the project.
3. Decide which methods and tools will be used for each stage of the project.
4. Decide how the process and outcomes will be documented.
5. Decide how the project will be evaluated.
6. Carry out each stage of the project, ensuring that the requirements match the design.
7. Carry out a continuous process of documentation and evaluation.

The process starts by specifying the *General Requirements* for the system to be designed. These provide broad yet precise initial requirements and constraints for the proposed system, in language that designers and customers can understand. They are used to guide the design and to provide a reference for validation of the system. The requirements should normally indicate:

– the scope of the project;
– the main actors involved in designing, deploying, using, maintaining the system;
– the market need and business case;
– general attributes and constraints of the proposed system (such as whether it aims to support individual or collaborative working).

The requirements will be extended and made more precise as the project progresses.

This leads to two parallel studies: a theory-based study of the underlying cognitive processes and social activities, and an investigation into how everyday activities are performed in their normal contexts. The *Theory of Use* involves an analysis of relevant literature from cognitive psychology, social sciences and business management, to form a rich picture of the human knowledge and activity. It is essential that this should offer a clear guide to system design. Thus, it must be relevant to the intended use of the system and extend the requirements in a form that can be interpreted by software designers and engineers.

The aim of carrying out *Field Studies* is to uncover how people interact with current technology in their normal contexts. The role of the fieldworker is both to interpret activity and to assist technology design and organizational change. This addresses the widely recognized problem of ethnographic approaches that, while they can provide an understanding of current work practices, they are not intended to explore the consequences of socio-technical change.

Table 1 shows a multi-level structure for field studies, with level 1 consisting of a survey of the existing organizational structures and schedules, levels 2 and 3 providing an analysis of situated practices and interactions of those for whom the technology is intended, and level 4 offering a synthesis of the findings in terms of designs for new socio-technical systems. The four levels give an overview of activity, leading to more detailed investigation of particular problem areas, with each level illuminating the situated practices, and also providing a set of issues to be addressed for the next level. These piece together into a composite picture of how people interact with technology in their everyday lives, the limitations of existing practices, and ways in which they could be improved by new technology.

Level 1	*Activity structures and schedules*
Activity:	Study work plans, organizational structures, syllabuses, resources.
Purpose:	To discover how the activities are supposed to be conducted.
Outcome:	Description of the existing organizational and workplace structures; identification of significant events.
Level 2	*Significant events*
Activity:	Observe representative formal and informal meetings and forms of communication.
Purpose:	To discover how activities, communication, and social interaction are conducted in practice.
Outcome:	A description and analysis of events that might be important to system design; identification of mismatches between how activity has been scheduled and how it is has been observed to happen.
Level 3	*Conceptions and conflicts*
Activity:	Conduct interviews with participants to discuss areas of activity needing support, breakdowns, issues, differences in conception.
Purpose:	To determine people's differing conceptions of their activity; uncover issues of concern in relation to new technology; explore mismatches between what is perceived to happen and what has been observed.
Outcome:	Issues in everyday life and interactions with existing technology that could be addressed by new technology and working practices.
Level 4	*Determining designs*
Activity:	Elicitation of requirements; design space mapping; formative evaluation of prototypes.
Purpose:	To develop new system designs.
Outcome:	Prototype technologies and recommendations for deployment.

Table 1. Multi-level structure for field studies

The outcomes of these two studies are synthesized into a *Task Model*. This is a synthesis of theory and practice related to how people perform relevant activities with their existing technologies. It is the least intuitive aspect of socio-cognitive engineering, and it is tempting to reduce it to a set of bullet-point issues, yet it provides a foundation for the systems design. It could indicate:

- the main actors and their activity systems;
- how the actors employ tools and resources to mediate their interaction and to externalise cognition;
- how the actors represent knowledge to themselves and others;
- the methods and techniques that the actors employ, including differences in approach and strategy;
- the contexts in which the activities occur;
- the implicit conventions and constraints that influence the activity;
- the actors' conceptions of their work, including sources of difficulty and breakdown in activity and their attitudes towards the introduction of new technology.

The *Design Concept* needs to be developed in relation to the Task Model. It should indicate how the activities identified by the Task Model could be transformed or enhanced with the new technology. It should:
- indicate how limitations from the Task Model will be addressed by new technology.
- outline a system image (Norman, 1986) for the new technology;
- show the look and feel of the proposed technology;
- indicate the contexts of use of the enhanced activity and technology;
- propose any further requirements that have been produced as a result of constructing the design concept.

The Design Concept should result in a set of detailed design requirements and options that can be explored through the design space.

The relationship between the Task Model and Design Concept provides the bridge to a cycle of iterative design that includes:
- generating a space of possible system designs, systematically exploring design option and justifying design decisions;
- specifying the functional and non-functional aspects of the system;
- implementing the system;
- deploying and maintaining the system.

	Software Engineering	Task Engineering	Knowledge Engineering	Organizational Engineering
Maintain	Installed system	New task structure	Augmented knowledge	New organizational structure
Evaluate	Debugging	Usability	Conceptual change, skill development	Organizational change
Integrate	Prototype System			
Implement	Prototypes, Documentation	Interfaces, Cognitive tools	Knowledge representation	Communications, Network resources
Design	Algorithms and heuristics	Human-computer interaction	Domain map, user model	Socio-technical system
Interpret	Task Model			
Analyze	Requirements	Tasks: goals, objects, methods	Knowledge: concepts, skills	Workplace: practices, interactions
Survey	Existing systems	Conventional task structures and processes	Domain knowledge	Organizational structures and schedules
Propose	General Requirements			

Table 2. A 'building block' framework for socio-cognitive system design.

Although these stages are based on a conventional process of interactive systems design (see (Preece, Rogers, & Sharp, 2002) for an overview), they give equal emphasis to cognitive and organizational factors as well as task and software specifications. The stages shown in Figure 1 are an aid to project planning, but are not sufficiently detailed to show all the design activities. Nor does the Figure make clear that to construct a successful integrated system requires the designers to integrate software engineering with design for human cognition, social interaction and organisational management. The 'building block' diagram in Table 2 gives a more detailed picture of the systems design process.

The four 'pillars' indicate the main processes of software, task, knowledge and organizational engineering. Each 'brick' in the diagram shows one outcome of a design stage, but it is not necessary to build systematically from the bottom up. A design team may work on one 'pillar', such as knowledge engineering, up to the stage of system requirements, or they may develop an early prototype based on a detailed task analysis but without a systematic approach to software engineering. How each activity is carried out depends on the particular application domain, actors and contexts of use.

The design activities are modular, allowing the designer to select one or more methods of conducting the activity, according to the problem and domain. For example, the usability evaluation could include an appropriate selection of general methods for assessing usability, or it could include an evaluation designed for the particular domain.

It should be emphasized is that the blocks are not fixed entities. As each level of the system is developed and deployed is will affect the levels below, (for example, building a prototype system may lead to revising the documentation or re-evaluating the human-computer interaction; deploying the system will create new activities). These changes need to be analysed and supported through a combination of new technology and new work practices. Thus, the building blocks must be revisited both individually to analyse and update the technology in use, and through a larger process of iterative re-design.

Although the Table 1 shows system evaluation as a distinct phase, there will also be a continual process of testing, to verify and validate the design, as shown on Figure 1. Testing is an integral part of the entire design process, and it is important to see it as a lifecycle process (Meek & Sharples, 2001), with the results of from testing of early designs and prototypes being passed forwards to provide an understanding of how to deploy and implement the system, and the outcomes of user trials being fed back to assist in fixing bugs and improving the design choices.

The result of the socio-cognitive engineering process is a new socio-technical system, consisting of new technology and its associated documentation and proposed methods of use. When this is deployed, in the workplace, home, or other location it should not only produce bugs and limitations that need to be addressed, but also engender new patterns of work and social and organizational structures which become contexts for further analysis and design.

FUTURE TRENDS

The computer and communications industries are starting to recognise the importance of adopting a human-centred approach to the design of new socio-technical systems. They are merging their existing engineering, business, industrial design and marketing methods into an integrated process, underpinned by rigorous techniques to capture requirements, define goals, predict costs, plan activities, specify designs and evaluate outcomes. IBM, for example, has developed the method of User Engineering to design for the "total user experience". (IBM, 2004). As web-based technology becomes embedded into everyday life, then it is will be increasingly important to understand and design distributed systems for which there are no clear boundaries between people and technology.

CONCLUSION

Socio-cognitive engineering forms part of an historic progression from user-centred design and soft systems analysis towards a comprehensive and rigorous process of socio-technical systems design and evaluation. It has been applied through a broad range of projects for innovative human technology and is still being developed, most recently as part of the European MOBIlearn project.

REFERENCES

Checkland, P., & Scholes, J. (1990). *Soft Systems Methodology in Action*. Chichester: John Wiley and Sons.

Greenbaum, J., & Kyng, M. (Eds.). (1991). *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

IBM. (2004). *User Engineering*. Retrieved 12th August, 2004, from the World Wide Web: http://www-306.ibm.com/ibm/easy/eou_ext.nsf/publish/1996

Meek, J., & Sharples, M. (2001). A Lifecycle Approach to the Evaluation of Learning Technology, *Proceedings of CAL 2001 Conference, University of Warwick, UK* (pp. 195-196).

Mumford, E. (1995). *Effective Systems Design and Requirements Analysis: The ETHICS Approach*. Basingstoke: Macmillan.

Norman, D. A. (1986). Cognitive Engineering. In D. A. Norman & S. W. Draper (Eds.), *User Centred System Design*. Hillsdale, New Jersey: Lawrence Erlbaum.

Norman, D. A., & Draper, S. (1986). *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction Design: Beyond Human-Computer Interaction*. New York, NY: John Wiley & Sons.

Rogers, Y., & Bellotti, V. (1997). Grounding blue-sky research: How can ethnography help? *Interactions, May - June 1997,* 58-63.

Sachs, P. (1995). Transforming work: collaboration, learning and design. *Communications of the ACM, 38*(9), 36-44.

Sharples, M., Corlett, D., & Westmancott, O. (2002). The Design and Implementation of a Mobile Learning Resource. *Personal and Ubiquitous Computing, 6*, 220-234.

Sharples, M., Goodlet, J., & Pemberton, L. (1992). Developing a Writer s Assistant. In J. Hartley (Ed.), *Technology and Writing: Readings in the Psychology of Written Communication* (pp. 209–220). London: Jessica Kingsley.

Sharples, M., Jeffery, N., du Boulay, J. B. H., Teather, D., Teather, B., & du Boulay, G. H. (2002). Socio-Cognitive Engineering: A Methodology for the Design of Human-Centred Technology. *European Journal of Operational Research, 132*(2), 310-323.

Sharples, M., Jeffery, N. P., du Boulay, B., Teather, B. A., Teather, D., & du Boulay, G. H. (2000). Structured Computer-based Training and Decision Support in the Interpretation of Neuroradiological Images. *International Journal of Medical Informatics, 60.*(30), 263-228.

TERMS AND DEFINITIONS

Activity system. The assembly and interaction of people and artefacts, considered as a holistic system that performs purposeful activities. See http://www.edu.helsinki.fi/activity/pages/chatanddwr/activitysystem/

Human-centred design. The process of designing socio-technical systems (people in interaction with technology) based on analysis of how people think, learn, perceive, work and interact.

Socio-technical system. A system comprising people and their interactions with technology, for example the worldwide web.

Soft systems methodology. An approach developed by Peter Checkland to analyse complex problem situations containing social, organisational and political activities.

System image. A term coined by Don Norman (Norman, 1986) to describe the guiding metaphor or model of the system that a designer presents to users (e.g. the 'desktop metaphor', or the telephone as a 'speaking tube'). The designer should aim to create a system image that is consistent, familiar, where possible, enables the user to make productive analogies.

Task analysis. An analysis of the actions and/or knowledge and thinking that a user performs to achieve a task. See http://www.usabilitynet.org/tools/taskanalysis.htm

User-centred design. A well-established process of designing technology that meets users' expectations, or that involves potential users in the design process.

User Engineering. A phrase used by IBM to describe an integrated process of developing products that "satisfy and delight users".