The Nature of User Interface Design – The Role of Domain Knowledge

JAN GULLIKSEN[1,2] (jg@syscon.uu.se),
BENGT SANDBLAD[1,2] (bengt.sandblad@cmd.uu.se) &
MATS LIND[1] (mats.lind@cmd.uu.se)

[1] Uppsala University, Center for Human-Computer Studies, Lägerhyddvägen 18, S-752 37 Uppsala, Sweden.
[2] Uppsala University, Department of Technology, Systems & Control Group, PO Box 27, S-751 03 Uppsala, Sweden.

Abstract

The importance and growing awareness of domain knowledge acquisition in information systems development and, especially, in the process of design of human-computer interfaces, are becoming more evident. In several in-house development projects, methods for efficiently capturing and utilising domain knowledge have been defined and tested (e.g., through the definition of domain specific style guides and analysis of information utilisation). Also, methods for modelling case handling work in general terms, which is the main application focus of our studies, have been developed and tested in larger organisations.

This paper focuses on the relation between domain knowledge models and conceptual models in information system development and user models in human-computer interaction. Based on Norman's model of user perception of an existing computer system, different limitations are identified and possible extensions are discussed. By introducing the domain context, several additional models of the work task by the user, as well as the designer, occur. If the dynamic iterative system development process is regarded, additional models and dynamic changes in the models over time can be traced. Furthermore, these models are related to the state-of-the-art knowledge on mental models and domain modelling. Implications for design, such as work modelling, iterative system development, analysis of information utilisation, design and evaluation methods, are discussed.

Keywords

Mental Models; Design Methods; Domain Knowledge; Iterative System Development

1. INTRODUCTION

Domain modelling and domain knowledge acquisition methodologies have received increasing attention lately within human-computer interaction. The understanding of the task domain is particularly important for the design of efficient human-computer interfaces. The natural work, time, motivational and social context is essential, if not crucial, for the understanding of the work domain. This is why a context sensitive approach is needed [Whiteside, Bennett, & Holtzblatt, 1988].

The extension and limitation of what we call a domain is particularly interesting. The shifting foci on interface design from machine-ware, through single-user human-computer interfaces to the socio-technical environment [Grudin, 1990], suggests that the domain should never be limited in a way that factors that could influence the performance of work with computer support risk being left out. Design guidelines, such as the striving for consistency have, in various situations, been shown to attract attention away from the proper focus of user interface design, namely the user, the work task and the understanding of the work domain [Grudin, 1989].

It is important to sense the differences in various development contexts such as in-house development, commercial products development and competitively bid contract projects [Grudin, 1991]. In contract development, users are known from the outset, but the development team is identified only after a contract has been awarded. In product development, developers are known from the outset but users can merely be identified after the product has reached the market. In in-house development both users and developers are known from the outset. This implies that one of the prerequisites for an efficient development team – good representation from the domain – is fulfilled.

An early focus on users, through the use of interactive design, empirical measurement and iterative design to capture the domain knowledge is essential [Gould & Lewis, 1985]. By introducing a "spiral" model of system development more efficient user interfaces can be designed [Boehm, 1988]. Essential in this model are succeeding iterations to capture, interpret and refine the representations of domain knowledge, using prototyping techniques and user involvement. One framework for this view on system development is the task-artefact cycle [Carroll, 1991]. The purpose is to better understand the tasks people are undertaking, and to apply this understanding in the design process. An important part of this understanding concerns the psychology of task behaviour. The task-artefact cycle approach also includes the fact that the use of an artefact in a work environment will redefine the tasks for which the artefact was originally designed.

1.1 Domain Modelling and Domain-specific Design

Recent research within cognitive ergonomics stresses the importance of extending our user interface horizon beyond the visible user interface to designing users' interactions with domains [Fischer, 1993]. Knowledge about the importance of domain modelling date back as far as Gibson [1977] who, among others, stressed that psychology should be the study of the interaction between human beings and their environment. We are, as human beings, able to directly perceive the environment's constraints and affordances (possibilities), which are necessary in order to be able to achieve our goals. Within the cognitive engineering approach, it is noted that the human is not a passive user of a computer program, but an active problem solver in some world [Woods & Roth, 1988]. In Rasmussen & Vicente [1990], the importance of the work task as a fundamental unit in the human-machine system is emphasised. Relating it to ecological psychology, domain analysis could be described in terms of its affordances and constraints and the goals of the user as states of the domain [Vicente, 1990]. Information that the perceivers actually are using in their work is important to attain a valid domain model [Neisser, 1987]. Domain modelling can also be useful for software engineering and should have its greatest impacts on software reuse [Arango & Prieto-Diaz, 1991].

In Gulliksen [1996], case handling models have been discussed in terms of a general framework for capturing domain knowledge. These work activity specific domain models have proven to be not only essential, but a necessary prerequisite for, in order to

be able to create and design domain specific user interfaces. These domain model structures are dynamic, which is necessary to meet the coming changes in the work activities of the organisation, and also for being able to compromise between different demands from these work activities. The domain models are based on the assumption that technological changes in the organisation can not be made without meeting the necessary changes in the organisational structure, the human being and his/her competence, and the work activity as such [Leavitt, 1958].

Domain-specific design and domain-specific style guides [e.g., Gulliksen & Sandblad, 1995a] are methods that have received extensive attention in later years in larger corporations for their development strategies. By introducing high level standards that contain domain knowledge, development of applications can be made faster, cheaper and easier. This can also lead to increased possibilities for user participation and result in more usable user interfaces. We consider this a very important method for capturing domain knowledge.

1.2 User Centred System Design – Cognitive Engineering

To be able to understand the system development process we need to extend the well known model of user perception of an existing computer system (Figure 1).

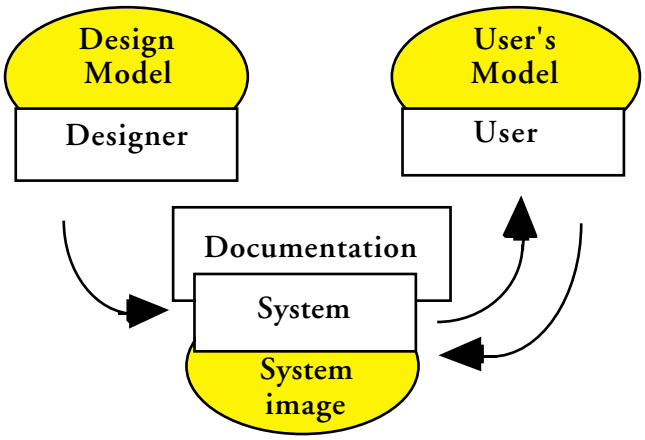

Figure 1. The System Image, the resulting abstraction of the designer's conceptual model, is what the user interacts with to establish a user's mental model of the system. (From Norman, D.A., 1986, Cognitive Engineering, In User Centered System Design, Norman & Draper (eds.) Lawrence Erlbaum Associates.)

Norman [1986] identifies the **Design Model** – that is, the conceptual model of an information system held by the designer, the **User's Model** – that is the conceptual model formed by the user and the **System Image** – the interface image resulting from the physical structure that has been built (including documentation and instructions). The System Image is the physical image of the computerised work situation. By perceiving this, the user's mental model can be derived. The design problem is to create a system that follows a consistent, coherent conceptualisation (the design model) so that the user can develop a mental model (user model) of that system consistent with the design model. Note that the User's Model is not formed from the design model but from the

way the user interprets the System Image. It should be realised that everything the user interacts with helps to form that image.

2. A STRUCTURE OF MENTAL MODELS IN EXPERIMENTAL ITERATIVE SYSTEM DEVELOPMENT

The model should, however, be extended for understanding the development and refinement of mental models *during* the process of system development. First, this model does not regard the work task to be accomplished with the support and, second, the model does not capture aspects stemming from the iterative task and system development process.

2.1 Possible Extensions: The Work Task

Hence, it is necessary to extend Norman's model with the actual work task for which the user wants, or needs, in a computer system (Figure 2).

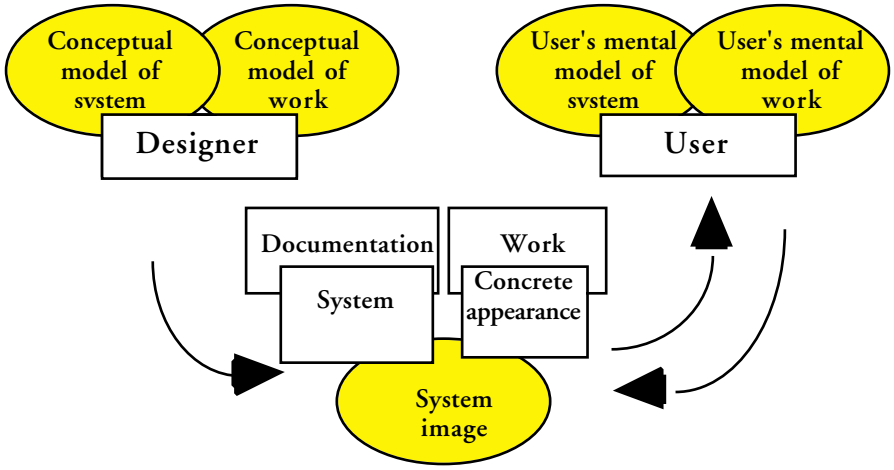

Figure 2. By introducing a basic work task in Norman's view of the mental models involved in the perception and interpretation of a computerised work situation it will be important to establish additional mental models of the work task by both the designer and the user.

A "basic task" is to be accomplished what ever system or user interface the user is using. Examples of such tasks are: the writing of a book, to decide whether or not to buy stocks of a certain price and kind, to see to it that someone's wage gets paid, etc.

Performing a basic task is the utmost goal of a professional user's interaction with the system. It is our view that the designer of the functionality of the software must have a thorough knowledge of certain aspects of the work task to be able to design an effective user interface. The ideal situation would be if the designer's mental model of work could be similar to the user's mental model. The best prerequisite for design would, perhaps, have been a formal description of the user's mental model of work. Unfortunately, capturing of mental models is more or less impossible. Work is an abstract concept that the designer can only view through its observational methods such as inter-

viewing and observing a worker interacting with his/her work task and by studying existing physical descriptions of the goals of the work activity. Of course, the actual user is the expert in his/her own area of work, which is why participatory design techniques should be appropriate [Schuler & Namioka, 1993; Greenbaum & Kyng, 1991]. On the other hand, if adopting more user controlled design methodologies [Bjerknes, Ehn, & Kyng, 1987; Ehn, 1988], the limitations of the techniques become more evident. Users are, although necessary and essential in the development work, not in anyway experts in human-computer interaction design. Neither do they have the ability to critically analyse their work and establish automatic behaviour patterns nor possible shortcuts. That is, a designer needs to be introduced into the development work with the issue of establishing appropriate mental models of certain aspects of the basic tasks and to be able to design well-functioning user interfaces. It is a somewhat different system image that occurs when the designers incorporate their conceptual model of work into a conceptual design model of a system. When this design model is implemented in the system the system image that that system creates is task supporting in that particular domain. These mental models held by the designer have to be formed somehow.

2.2 Possible Extensions: Iterative Experimental System Development

Norman's model (Figure 1) is a static description of mental models involved in human-computer interaction. Obviously, mental models evolve and change over time, just as work patterns change as the user acquires expertise. An understanding of the modelling of the domain knowledge requires an understanding these mental models that are created and changed in iterative experimental system development. We analyse skilled routine workers that are experts in their domain of work, but these workers are not in any sense computer experts. Their view of their work has derived from a long period of adaptation and specialisation. Work can be complicated and cognitively demanding at the same time. This is because work contains automatisable routine micro tasks that can be performed simultaneously and in parallel with a cognitively demanding task. The work can include interaction with clients or fellow workers, telephone calls, meetings, calculations, information collection, decision making and reading and writing in different combinations. This type of work usually requires a great extent of task switching [Bannon, Cypher, Greenspan, & Monty, 1983; Henderson & Card, 1987].

In general, the professional worker is the best expert in his/her area of work and hence can perform the work in the best way. Before the computerisation project, the user has a functioning model of the work situation with which he/she interacts (Figure 3).

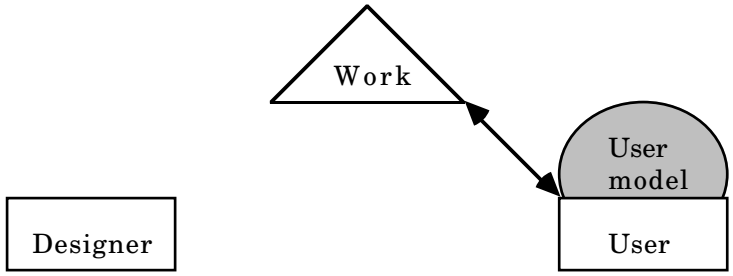

Figure 3. The work situation before starting the computerisation project.

If a computerisation of a work situation is decided upon, the user interface designer has to analyse and collect samples from the work situation based on, for example, an analysis of information utilisation [Gulliksen, Lif, Lind, Nygren, & Sandblad, 1996]. The information needed in the different work steps can be captured, and observations can be made on how this information is being physically manipulated. This is essential in the quest for cognitively demanding aspects of information use when designing computer support. The work can also be viewed according to the overall goals and expectations of the future work situation, independent of the aids for the current work situation (e.g., predicting whether a special operation is necessary to perform by a particular worker or if another competence in the organisation is required; predicting if a computer system automatically could perform an operation). The designer establishes a conceptual model of the user's work situation by analysing and observing the work as such and by interviewing potential end users (Figure 4).

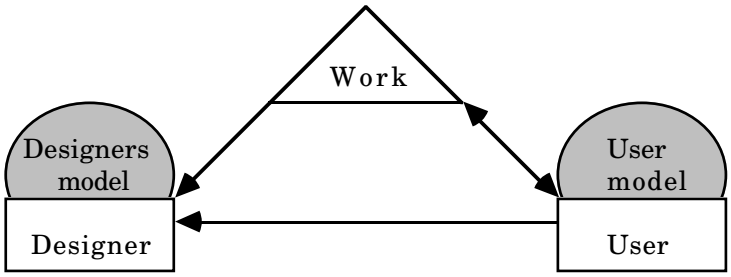

Figure 4. The analysis of information utilisation captures important aspects of information handling both from interviewing intended users and observing concrete appearances of the work situation.

Errors occur both as designers misinterpret the user's mental model when observing the user interact with the work situation, and because of the problems of capturing all aspects of the work without the designers themselves performing it. Difficulties in specifying the results of the analysis of information utilisation in a formal or semiformal language, can also cause errors, as well as aspects that can not be formally specified [Gulliksen et al., 1995].

According to this methodology the design process is then defined as a translation of a conceptual model of prospects of the work into a system model by the designer. Here, the designer has the possibility to capture aspects that could not be formally specified and present it in a design. The system model is the basis for the development process. The designer should have aids for describing an interface prototype so as not to be misinterpreted and sufficient competence to conduct a design that can be implemented. One should be aware of the fact that a computer's system image will bring changes to the entire user model and, therefore, could change the work routines dramatically. The designer's model of the users work is the prerequisite for the design on how work is to be performed and which aspects that change due to computerisation. The user can now establish a user's system model by using the system image in the work context, and then matching it with the simplest possible mental model. (Figure 5).

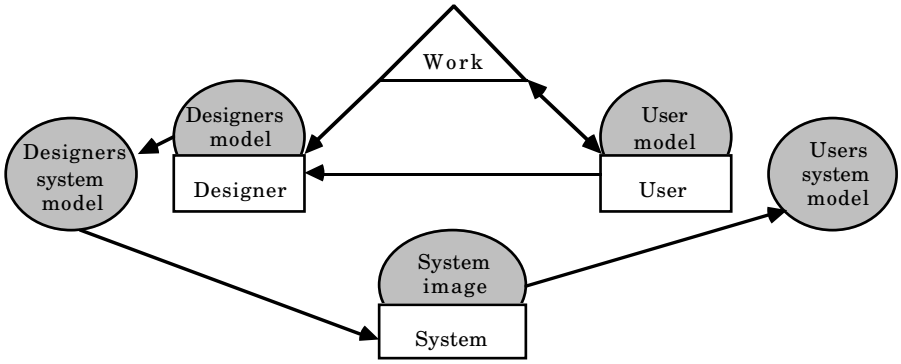

Figure 5. The designer has to translate the design model into a system model to be able to create a system for the user to perceive and learn to operate by establishing a user's system model.

This design process should be performed by a designer not directly involved in the work himself, but with much experience in interface design and with a set of basic interface design heuristics available (e.g., how much information to present on the screen simultaneously, colour coding, pattern recognition possibilities, automatisable processes, visual coding, etc.). The user's system model can implicate new views on the work tasks and therefore revised mental models. This system might be a prototype in the first iteration and will further change as the system develops. An information system is never actually brought to completion, but continuously undergoes revision and maintenance as the work changes and develops. Therefore, the model continues in a spiral process (Figure 6).

It is important to keep down the iterative system development cycle time. It is also important that the person performing the analysis is the same person as the one performing the design to obtain the best communication of informal results of the analysis. If development could be performed based on the above model, simultaneous development of the information system, the work task, the user and his/her competence and the organisational setting in which this occurs can take place. This would then lead to better (more usable) user interfaces, better utilisation of the development resources and greater satisfaction by all participators in the development process.

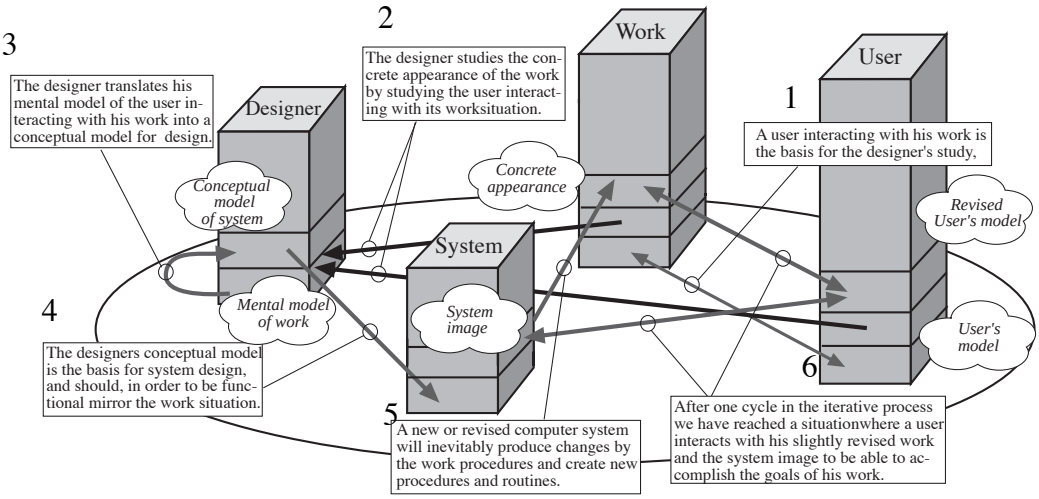

Figure 6. One iterative cycle in the experimental development process of information systems. The process should be read clockwise illustrating required work processes and mental models.

2.3 Presumed Gain from Extending Norman Theories

To understand what mental models that are formed in user centred system development, how they are formed, and the importance of extending the task domain is an important prerequisite for understanding and acquiring the relevant domain knowledge for successful user interface design. Field studies on in-house development projects have shown numerous examples of where domain studies have been deliberately ignored, improperly communicated or distorted due to limitations in development tools and limited opportunities for effective user involvement. It is our belief that the theories above can support a knowledge basis for system development. This involves the establishment of appropriate methods for analysis and documentation of analysis results, for communication of formal and informal domain knowledge, for development of design methods and tools that has domain knowledge acquisition as a substantial part. For example, the method of analysis of information utilisation [Gulliksen et al., 1996], that especially focuses on factors affecting cognitive load or domain specific design methodologies with domain specific style guides are frameworks for the establishment of domain models..

3. ITERATIVE EXPERIMENTAL SYSTEM DEVELOPMENT

There is an obvious parallel between the capturing of mental models and the formulation of requirement specifications in a development project. There is also a parallel between the development and evaluation of prototypes on the one hand, and the user's ability to understand how the work and work environment will be effected by the introduction of new computer artefacts on the other. To support efficient development of such models, the procedures and techniques for participatory design and user centred system development must be well adapted to this purpose. In this context, the use of domain specific methods and techniques are especially important. With such methods, it might, for example, be possible to specify requirements etc. directly in domain terminology; that is, in the language of the user.

Participatory design means that representatives from the work domain (i.e. the organisations into which the system under development is to be implemented) are involved in the development process. Domain representatives (normally referred to as "end users"), are those interested in the formulation of requirement specifications or the evaluation of prototypes in an organisation. Other representatives from the organisation should be involved, as well. Participatory design should include (at least) three different participants: domain experts, designers and developers. The designers are responsible for design and evaluation of system functionality and user interface. The developers are responsible for the implementation of the designed system and interface. In practise, the roles of designers and developers are often performed by the same person. It is, however, important to distinguish between these different roles of work. A basic principle is that the domain experts should only specify their requirements in work activity related terms, and the role of the designer is to bridge the gap between domain experts and developers. The communication between these different participants are based on their mental and conceptual models during the phases of the development process.

There are many obstacles to overcome if participatory design is to be efficient. The involved persons must be mighty competent and skilful. The organisation of the development project must be efficient, especially concerning how much resources should be allocated to the domain experts so that they can contribute in an efficient way. More

importantly is the support given for how knowledge on the mental and conceptual models can be incorporated in the design process.

There are different approaches to how the user centred development process is organised and to which aspects that are considered. The overall purpose must be to base the development model on such basic requirements as discussed above. Some important aspects concerning aspects to be considered in a user centred development model are discussed in the task-artefact model [Carroll, 1991]. The task-artefact model includes a psychological approach to understanding users and their tasks, together with the observation that there is a strong feed-back from the artefact to the performance of tasks. The method for analysis of information utilisation, as discussed above, is one method for analysing cognitive aspects of how users perform their tasks in a specific work environment. The feed-back aspect, i.e. how the use of an artefact can redefine the way it is performed, can only be treated by active user involvement in an iterative process.

Prototyping is an important technique in participatory design. In order to efficiently support the design process, prototypes must visualise all relevant aspects of the artefact under development. Prototypes can be seen as tools for making models concrete, and can facilitate communication of models between the participants involved in the participatory development. If prototyping, as a design support method, is to be efficient, the quality of the prototyping tools is essential. An efficient prototyping tools must support the cooperative process. It must have the capabilities to rapidly produce new versions for usability tests. The continuous implementation of new domain specific interface elements as parts of the tool must also be supported. Today's tools do not fulfil these requirements. Further research and development is therefore necessary.

In a user centred design process there is a need for appropriate evaluation methods and techniques. Prototypes must be evaluated with regard to utility and usability criteria [Nielsen, 1993; Lif & Sandblad, 1996]. Evaluation methods will, in this way, have the function of verification of the user's mental model of the system and the designer's conceptual models.

4. SPECIAL REQUIREMENTS FOR SKILLED USERS

For skilled professionals, efficiency is necessary concerning both the interface in a work environment and the development process. An interface is efficient in the work process for end users if it is cost effective, has the right functionality and is "obvious" to the user [Nygren, Johnson, Lind & Sandblad, 1992]. This means that the user can devote most of his concentration on the work process and spend a minimum of cognitive effort handling the interface.

Most administrative routine work involves making judgements and decisions. Thus, computer systems designed to support human work in such domains are systems that support decision making. Decision making is a demanding cognitive process and human cognitive abilities for decision making have limited capacity. Decision making and the control of the human-computer interface should be regarded as two concurrent tasks competing for the cognitive resources of working memory [Lind, 1991]. The main objective for construction of user interfaces should be to make the control of the user interface possible to perform on an automatic cognitive level, leaving the high level cognitive capacity for the decision making process. This can be accomplished by designing the interface so that it requires a minimum amount of attention from the decision maker during the decision period (e.g. by avoiding scrolling or paging, avoiding the need to call up, re-size and move additional windows or replying to modal dialogue boxes). We, therefore, need to identify major decision making situations and the

information needed in these situations to be able to pre-determine a specific layout and functionality for each type of decision making situation.

Problems arise when a user is attempting to accomplish several different tasks in a single session [Bannon, Cypher, Greenspan, & Monty, 1983], which is very common in case handling work. The concept of 'workspaces' is introduced as an interface metaphor that supports activity coordination and, thereby, reducing mental workload.

Development of new computer systems is often a basis for the development of the entire work process in an organisation. It should be possible to treat organisational aspects and information handling aspects parallel to each other. Therefore, traditional methods for work analysis need to be revised. Expectation analysis of workers in an organisation can be used to dictate goals for the work development. Goal-expectation conflicts are often sources to low user acceptance, but could perhaps be solved if recovered early.

5. DISCUSSION

The problems that results from the separation of the interactive software functionality and its form (symbolised by the human-computer interface), are enhanced according to the different research communities – one focusing on information system functionality and organisational impacts, and one focusing on human-computer dialogue, or user interfaces that are engaged in the development [Grudin, 1992]. Recently, an attempt to bridge the gap between observation of end users' needs and the support for system design by a functional information and knowledge acquisition modelling method for capturing domain knowledge has been made [Sundström & Salvador, 1995]. It stresses that a user-centred design approach is incomplete unless it incorporates techniques for analysing the operational environment. The need for incorporating domain knowledge in the software development process becomes more apparent and possibilities for it are also increasing, due to the illumination of the problem of lacking domain knowledge acquisition.

The theories stressed in this paper constitute a theoretical framework for several methodologies for information technology and domain development. Existing methods for task analysis need to be extended with a method that we call **analysis of information utilisation** (AIU). This method focuses on how information entities encountered in the information analysis are physically being manipulated, and especially factors concerning cognitive load. There is a need for **domain specific design methodologies** that facilitate the incorporation of domain knowledge early in the design process. These are described in Gulliksen & Sandblad [1995a]; Borälv et al. [1994]; Gulliksen et al. [1993]; Olsson et al. [1993]; Gulliksen & Sandblad [1995b]. Research on general domain modelling methodologies has only recently been initialised. So far, this research has been successful within the case handling framework [Gulliksen, 1996].

In the future, we will continue to work on specifying these development aids to be able to incorporate domain knowledge into every step of the interactive system development process.

6. REFERENCES

ARANGO, G. & PRIETO-DIAZ, R. (1991). Introduction and Overview: Domain Analysis and Research Directions. *In Prieto-Diaz, R. & Arango, G. (eds.) Domain Analysis and Software Systems Modelling,* IEEE Computer Society Press, California.

BANNON, L., CYPHER, A., GREENSPAN, S. & MONTY, M.L. (1983). Evaluation and Analysis of Users' Activity Organisation. *In A. Janda (ed.) Human factors in computing systems. Proceedings of CHI '83,* North-Holland, Amsterdam. pp 54-57.

BJERKNES, G., EHN, P. & KYNG, M. (1987). *Computers and Democracy.* Gower Publishing Company Ltd., England.

BOEHM, B. (1988). A Spiral Model of Software Development and Enhancement. *Computer, vol. 21, No .5,* pp. 61-72.

BORÄLV, E., GÖRANSSON, B., OLSSON, E. & SANDBLAD, B. (1994). Usability and efficiency. The HELIOS approach to development of user interfaces. *In U. Engelmann, F. C. Jean, P. Degoulet (Eds.) The HELIOS Software Engineering Environment, Supplement to Computer Methods and Programs in Biomedicine, Vol. 45*, pp. 47-64.

CARROLL, J.M., KELLOGG, W.A., & ROSSON, M.B. (1991) The Task-Artifact Cycle. *In J.M. Carroll (ed.) Designing Interaction. Psychology at the Human-Computer Interface* Cambridge University Press: Cambridge.

EHN, P. (1988). *Work-Oriented Design of Computer Artifacts.* Arbetslivscentrum, Sweden.

FISCHER, G. (1993). Beyond Human-Computer Interaction: Designing Useful and Usable Computational Environments. *In J. Alty, D. Diaper & S. Guest (eds.) People and Computers VIII, Proceedings of the BCSHCI '93 conference,* Cambridge University Press, Cambridge, UK

GIBSON, J.J. (1977). The Theory of Affordances. *In R.E. Shaw & J. Branford (eds.), Perceiving, Acting and Knowing: Toward an Ecological Psychology.* New Jersey: Erlbaum.

GOULD, J.D. & LEWIS, C. (1985). Designing for Usability: Key Principles and What Designers Think. *Communications. ACM, Vol. 28, No. 3,* pp. 300-311.

GREENBAUM, J. & KYNG, M. (Eds.) (1991). *Design at Work: Cooperative Design of Computer Systems.* Hillsdale, NJ: Lawrence Erlbaum Associates.

GRUDIN, J. (1989). The Case Against User Interface Consistency. *Communications. ACM, Vol. 32, No. 10,* pp. 1164-1173.

GRUDIN, J. (1990). The Computer Reaches Out: The Historical Continuity of Interface Design. *In J.C. Chew & J. Whiteside (eds.) Proceedings of Human Factors in Computing Systems, CHI '90, ACM/SIGCHI.*

GRUDIN, J. (1991). Interactive Systems: Bridging the Gaps Between Developers and Users. *IEEE Computer, Vol. 24, No. 4,* pp. 59-69.

GRUDIN, J. (1992). Utility and Usability: Research Issues and Development Contexts. *Interacting with computers, vol. 4, no 2,* pp. 209-217

GULLIKSEN, J., SANDBLAD, B., JOHNSON, M., LIND, M. & NYGREN, E. (1993). The Need for New Application Specific Interface Elements. *In G. Salvendy & M. J. Smith*

(eds.) Human-Computer Interaction, Proceedings of the 5th International Conference on Human-Computer Interaction, HCI International '93, Orlando, Florida, U.S.A, 8-13 August, 1993, Elsevier, pp.15-20

GULLIKSEN, J. (1996). Case Handling Models as a Basis for Information System Design. *In C.A. Ntuen & E.H. Park (eds.) Human Interaction with Complex Systems-II,* Kluwer Academic Publishers, Norwell, MA.

GULLIKSEN, J. & SANDBLAD, B. (1995a). Domain Specific Design of User Interfaces. *International Journal of Human-Computer Interaction, Vol. 7, No. 2,* pp. 135-151, Ablex Publishing Corporation, Norwood, New Jersey.

GULLIKSEN, J. & SANDBLAD, B. (1995b). Domain Specific Design of User Interfaces – Case Handling and Data Entry Problems. *In D. Benyon & P. Palanque (eds.) Critical Issues in User Interface Systems Engineering,* Springer Verlag.

GULLIKSEN, J., LIND, M., LIF, M. & SANDBLAD, B. (1995). Efficient Development of Organisations and Information Technology – A Design Approach. *In Y. Anzai and K. Ogawa (eds.) Symbiosis of Human and Artifact. Proceedings of the 6th International Conference on Human-Computer Interaction, Pacifico Yokohama, Yokohama, Japan 9 - 14 July 1995.*

HENDERSON JR., D.A. & CARD, S.K. (1986). Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface. *ACM Transactions on Graphics, vol. 5, No. 3,* pp. 211-243.

LEAVITT, H.J. (1958). *Managerial Psychology.* University of Chicago Press, Ltd. London.

LIF, M., & SANDBLAD, B. (1996). Domain-specific Evaluation, during the Design of Human-Computer Interfaces. *In A.G. Sutcliffe, F.Van Assche, & D. Benyon (eds.) Domain Knowledge for Interactive System Design. Proceedings of the IFIP WG 8.1/13.2 Joint Working Conference on Domain Knowledge for Interactive System Design, Geneva Switzerland, 8-10 May,* Chapman-Hall: London.

LIND, M. (1991). Effects of Sequential and Simultaneous Presentations of Information. *Report no. 19, CMD, Uppsala University.*

NEISSER, U. (1987). From Direct Perception to Conceptual Structure. *In U. Neisser, Concepts and Conceptual Development: Ecological and Intellectual Factors in Categorisation,* Cambridge University Press, Cambridge.

NIELSEN, J. (1993). *Usability Engineering.* Academic Press Inc. San Diego.

NORMAN, D.A. (1986). Cognitive Engineering. *In D.A. Norman & S. Draper (eds.) User Centered System Design,* Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey.

NYGREN, E., JOHNSON, M., LIND, M. & SANDBLAD, B. (1992). The Art of the Obvious. Automatically Processed Components of the Task of Reading Frequently Used Documents. Implications for Task Analysis and Interface Design. In J.P. Baursefeld, J. Bennett & G. Lynch (eds.) *Proceedings of Human Factors in Computing Systems, CHI '92,* Monterey, California, May 1992, ACM, pp. 235-239.

OLSSON, E., GÖRANSSON, B., BORÄLV, E. & SANDBLAD, B. (1993). Domain Specific Style Guide – Design and Implementation. *In proceedings of the MOTIF '93 & COSE International User Conference, Washington, D.C.*

RASMUSSEN, J. & VICENTE, K. (1990). Ecological Interfaces: a Technical Imperative in High-Tech Systems? *International Journal of Human-Computer Interaction, Vol. 2, No. 2,* pp. 93-111.

SCHULER, D., & NAMIOKA, A. (eds.) (1993). *Participatory Design: Principles and Practices.* Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey.

SUNDSTRÖM, G. & SALVADOR, A.C. (1995). Integrating Field Work in System Design: A Methodology and Two Case Studies. *IEEE Transactions on Systems, Man and Cybernetics, vol. 25, no. 3,* pp. 385-399.

WHITESIDE, J., BENNETT, J. & HOLTZBLATT, K. (1988). Usability Engineering: Our Experience and Evolution. *In M. Helander (ed.) Handbook of Human-Computer Interaction,* Elsevier: North-Holland.

VICENTE, K. (1990). A few implications of an ecological approach to human factors. *Human Factors Society Bulletin, Vol. 33, No. 11,* pp. 1-4.

WOODS, D.D. & ROTH, E.M. (1988). Cognitive Systems Engineering. *In M. Helander, (ed.) Handbook of Human-Computer Interaction.* Elsevier: North-Holland.

APPENDIX

A. IMPORTANT ASPECTS OF WORK

The types of work we mainly study is administrative case handling performed by *skilled professional* who only use and appreciate an information system that efficiently supports the main purpose of the task (e.g., to perform case handling). What a system and interface designer can see and study is the concrete appearance of the work situation by performing observation-interviews on presumptive end users, and to study the overruling goals of the work, as specified in the work activity plan. The designer can not capture the users' mental models in other ways.

Of particular importance is the view on work and the development of the work situation. Traditional methods for specification of user demands need to be seen from a perspective of the total domain of the work to be useful in the system development process. A "computerisation" of a work situation may not always be the right solution for an efficient work performance. The organisation of the work and its information handling routines are often closely related. Therefore, it is important to analyse the participators and the goals of the work. The development of computer systems needs to be a continuous process of smaller development projects with the purpose of achieving an organisation that can develop and "learn" through an experimental development process. This also puts demands on the competence of the workers and on the development tools and methods. Work is more than the description of work; it is the adaptation of individual work patterns for a specific work task. The system's functionality must mirror the mental models of the work to constitute efficient support. In analogy with the hammer, it does not matter how much effort is put into designing a hammer if the work task to be accomplished is sawing. The tool for aiding the performance of a task must not only be efficient but the right tool for the task.

According to the GOMS-modelling and the KLM-methods [Card, Moran & Newell, 1983], the speed of the interaction with the computer can be predicted with good precision. However, this concerns a trivial task. Field studies show that professional interaction with the computer in a real-life work setting of case handling means a few required key-pressings in longer periods of information search, judgement and decision processes. A typical period of 15 minutes of computer supported case handling work meant 4-5 keystrokes. This is we must conclude that working with computer support is so much more than just keyboard operations.

A.1 Judgements and Decision Making

It is important to distinguish between judgement and decision making as different tasks in the work of an administrative case administrator. *Judgement* is "the mental or intellectual process of forming an opinion or evaluation of discerning and comparing" and the capacity for judging is "the power or ability to decide on the basis of evidence". On the other hand a *decision* is "the act of settling or terminating...by giving judgement" [Webster's 3rd New International Dictionary]. Based on these definitions it could be deduced that the judgement is the process immediately preceding the momentary decision. Therefore it is relevant to speak about a decision point and a judgement process.

Furthermore, decision analysis involves an a priori decomposition of the decision process and a judgement analysis involves an a posteriori decomposition of the judgement process [Arkes & Hammond, 1986]. Here judgement is defined as a cognitive or intellectual process in which a person draws a conclusion, or an inference about some-

thing that cannot be seen on the basis of data that can be seen. Also, the level of difficulty in the tasks that the users are to face depend heavily on the complexity of the judgements they are to make. This is general knowledge on decision making and judgements, but all the more true when considering interactive work with the computer that involves decision making and judgements, in addition to the disturbance the computer media causes merely due to its existence.

B. HOW KNOWLEDGE ON MENTAL MODELS CAN BE USED IN INTERFACE DESIGN

In the early days of computing, the user of a computer system was the developer himself; he knew his needs and possibilities and had no problem establishing a mental model of the functionality of the computer system. The concept of **mental models** is borrowed from cognitive psychology and refers to a person's conceptualisation of a problem or a process.

B.1 Earlier Research on Mental Models

Mental models are established to be no more complicated than they need to be for explaining the consequence of the users actions [Johnson-Laird, 1983]. Mental models are considered to be a method for representing knowledge and the manipulation of models a form of reasoning. As a result of this, our mental models are incomplete, unstable, without firm boundaries, unscientific and parsimonious [Norman, 1983]. It is also important to distinguish between the **conceptual model**, devised as a tool for understanding and teaching physical systems and the **mental model**, which is what people really have in their heads and what guides their use of things. Too often there is no correspondence between the conceptual model of a system that guides the designer, and the system image that is created to produce the user's mental model of the system.

The concept of mental models is used frequently in the area of user interface design [Staggers & Norcio, 1993]. The relation between properties of the user interface and the ease by which a user forms a mental model of that interface has especially been considered. In DeKleer & Brown [1983], a distinction is made between **component models**, which concern the process of determining the function of a device by describing the behaviour of various components independent of the context in which the component is embedded, and **causal models**, as the end result of trial runs of the component model. These concepts concern how the mental models are used. It is known that users perform better when they are given a conceptual model before using the system [Staggers & Norcio, 1993; Carroll & Olson, 1988]. The use of analogies or metaphor functions as tools of thought have helped users to structure unfamiliar domains [Gentner & Gentner, 1983]. These structural relations between old and new areas are deemed identical as **structure-mapping**.

B.2 Structure and Contents of Mental Models

Many researchers consider mental models as organised structures of objects and their relationships [Gentner & Gentner, 1983; Williams, Hollan, & Stevens, 1983]. But they can also contain abstract notions [Johnson-Laird, 1983]. The user's mental models of a large and complex system can be decomposed into smaller, homomorphic, independent-

ly functioning, subsystems [Moray, 1987], which should be easier to capture or model than a mental model of the entire system. Another important feature of mental models is that they are executable, meaning that you can run a model to validate hypotheses about the reality of the representing model [DiSessa, 1983].

One example of the use of the mental model of the user and the conceptual model for design is the Command Language Grammar [Moran, 1981], where the system is divided into a conceptual, a communication and a physical component. These models should be congruent. This has further been outlined in Norman [1986]. Designers must, therefore, create and present a clear design model for the user so the latter is able to create appropriate mental models. User errors can also be reduced if the designer can anticipate users' models [Janosky, Smith, & Hildreth, 1986]. This leads to the conclusion that the designer has to be aware of the user's mental model [Staggers & Norcio, 1993].

Literature on mental models in human-computer interaction mostly regard mental models of the system without regarding the work setting. However, it is our view that extending the area of interest to include the designer, the design process, and the work tasks to be performed by the user, and then to look at the various mental models involved, leads to a number of interesting results. There is a need to study the end users' mental models and to try to capture the routes that users encounter to control, manoeuvre and navigate through computer artefacts. Extensive research work has been conducted in establishing the nature of mental models in human-computer interaction. The contents of these models, however, remain unknown.

B.3 The Problems of Capturing Mental Models

As mentioned before, mental models have been found to be fragmentary, incomplete and unstable (people forget details of a system), with limited possibilities for the users to "run" them and without firm boundaries (similar devices and operations get confused with one another). They are also unscientific (due to people sticking to "superstitious" behaviour) and parsimonious (people are willing to trade-off extra physical effort to reduce the mental complexity) [Norman, 1983].

B.4 A Discussion on Mental Models in User Interface Development

We want to be sure that a computer system works efficiently in its environment. Forced sequencialisation of the work, such as reading without the use of automatically interpretable patterns, can disturb the mental processes. Therefore, we need to capture and model the mental processes of work by determining where they start and stop, and produce a list of the essential information to be able to perform the work processes. The goal is, then, an effort to make this process as efficient as possible by supplying essential and non-disturbing computer support. Mental models of the work are mainly used for establishing the functionality of the user interface. But mental models are also essential for design. It is not the model, but when (work-related) it is needed and what environment (information) is needed.

Existing methods for analysing and modelling work situations are seldom dedicated to the capturing of the users' mental model structure. This is not especially surprising owing to the dynamic patterns of these mental models. Because of this, the solution might be to try and not capture the mental models, but to provide sufficient information for the users to effectively be able to establish their own mental models of a system in a work setting.

Appendix – References

ARKES, H.R. & HAMMOND, K.R. (eds.) (1986). *Judgement and Decision Making: an Interdisciplinary Reader*. Cambridge University Press, London.

CARD, S. K., MORAN, T. P. & NEWELL, A. (1983). *The Psychology of Human-Computer Interaction.* Lawrence Erlbaum Associates Ltd., Publishers, Hillsdale, New Jersey.

CARROLL, J.M. & OLSON, J.R. (1988). Mental Models in Human-Computer Interaction. *In M. Helander (ed.) Handbook of Human-Computer Interaction,* Elsevier Science Publishers B.V. (North Holland)

DEKLEER, J. & BROWN, J.S. (1983). Assumptions and Ambiguities in Mechanistic Mental Models. *In D. Gentner & A.L. Stevens (eds.) Mental Models,* pp. 155-190. Hillsdale, New Jersey: Lawrence Erlbaum Associates Inc.

DISESSA, A.A. (1983). Phenomenology and the Evolution of Intuition. *In D. Gentner & A.L. Stevens (eds.) Mental Models,* pp. 15-34. Hillsdale, New Jersey: Lawrence Erlbaum Associates Inc.

GENTNER, D. & GENTNER, D.R. (1983). Flowing Waters or Teeming Crowds: Mental Models of Electricity. *In D. Gentner & A.L. Stevens (eds.) Mental Models,* pp. 99-130. Hillsdale, New Jersey: Lawrence Erlbaum Associates Inc.

JANOSKY, B., SMITH, P.J. & HILDRETH, C. (1986). On-line Library Catalog Systems: an Analysis of User Errors. *International Journal of Man-Machine Studies, Vol. 25,* pp. 573-592.

JOHNSON-LAIRD, P.N. (1983). *Mental Models: Toward a Cognitive Science of Language, Inference and Consciousness.* Cambridge: Cambridge University Press.

MORAN, T.P. (1981). The Command Language Grammar: A Representation for the User Interface of Interactive Computer Systems, *International Journal of Man-Machine Studies, vol. 15,* pp. 3-50.

MORAY, N. (1987). Intelligent Aids. Mental Models and the Theory of Machines. *International Journal of Man-Machine Studies, Vol. 27,* pp. 619-629.

NORMAN, D.A. (1983). Some Observations on Mental Models. *In D. Gentner & L. Stevens (Eds.), Mental Models,* pp. 7-14, Hillsdale, New Jersey: Lawrence Erlbaum Associates Inc.

STAGGERS, N. & NORCIO, A.F. (1993). Mental Models: Concepts for Human-Computer Interaction Research. *International Journal of Man-Machine Studies, Vol. 38,* pp. 587-605.

WILLIAMS, M.D., HOLLAN, J.D., & STEVENS, A.L. (1983). Human Reasoning about a Simple System. *In D. Gentner & L. Stevens (eds.), Mental Models.* pp. 131-154, Hillsdale, New Jersey: Lawrence Erlbaum Associates Inc.