Analysis of Information Utilization (AIU)

JAN GULLIKSEN

MAGNUS LIF

MATS LIND

ELSE NYGREN &

BENGT SANDBLAD

Center for Human-Computer Studies (CMD), Uppsala University, Sweden.

Abstract

Analysis of Information Utilization (AIU) is a method for describing and analyzing how information entities identified in information analysis are being used in the work situation. AIU aims at complementing existing methods for user interface design, by identifying additional requirements for human-computer interaction. The method focuses on aspects of computer supported work, related to cognitive load, aspects of which end users often not are explicitly aware of.

For skilled workers, in a professional work environment, the efficiency of the user interface is extremely important. We have earlier stressed that important criteria for design of user interfaces deals with making the interface "obvious" to the users, by minimizing the cognitive load associated with the handling of it.

AIU is performed through observation-interviews in which human-computer experts interview representative users about their work situation and observe physical information-handling routines. The analysis identifies work tasks in terms of judgments and decision-making situations, requirements concerning the tasks that may have to be performed simultaneously, important features and priorities of the information, actions the user can initiate, and so on. The method supports the interface designer with human-computer interaction requirements structured for a

workspace oriented design. It is integrated in a user-centered development model, and supports the simultaneous development of competence, organization, work activity and information technology.

The article describes the background and contents of the method, and how the results of the analysis are documented and used in the design process. A number of application projects have shown that AIU makes it possible to capture aspects of human work and information processing, that are important to the design of better interfaces. AIU is not a new method for system development but complements today's methods for task and information analysis with more design-relevant information. For illustration, an example is presented, describing interface design based on AIU in a system development project in the Swedish National Tax Board.

1. Introduction

The development projects we mainly are involved in are in-house projects; that is, internal information system development projects, initiated and performed in a company by the company's own development team. In contrast to product development, in which the users seldom are known until the product reaches the market, and contract development, in which the users are known from the outset and the developers are not known until a development contract has been awarded, in-house development projects have the advantage of having defined users and developers from the project outset (Grudin, 1991a). Existing in-house development methodologies are based mainly on early focus on users, empirical measurement, and iterative design (Gould & Lewis, 1985). Although important, this is not enough. The software process model, determining the stages of interface development, is often a waterfall model; that is, it demonstrates successive stages of operational plans, operational specifications, coding specifications, coding, parameter testing, assembly testing, shakedown and system evaluation, each producing rigorous documentation. A spiral model of continuous analysis, design, evaluation and redesign could help develop and maintain larger information systems (Boehm, 1988).

Task analysis (TA) is a general concept describing methods and techniques eliciting descriptions of what people do, representing those descriptions, predicting difficulties, and evaluating systems against usability or functional requirements. Other TA methods predict performance, measure system complexity, or measure learnability or the transfer of knowledge between systems. TA is generally concerned with what people do to get things done (Preece et al., 1994). One can identify hierarchical TA methods; that is, a graphical representation for decomposing high level tasks into subtasks and operations (or actions) based on a structured chart notation (Shepherd, 1989). Hierarchical TA involves an iterative process of identifying, categorizing, and breaking tasks down into subtasks and checking the accuracy. All of this is

performed in conversation with and observation of users doing work. It results in an accurate description of the steps required to complete a task. Cognitive task analysis (CTA; Johnson, 1992) is, on the other hand, a group of methods concerned with enhancing the design process through the application of cognitive theories. CTA captures some representation of the knowledge users must have to complete a task. One of the most well known CTA methods is the model human processor that focuses on the capturing of appearances of the user's mental model of a computer system by predicting and modeling the user's represen tation of the interaction, for example, through goals, operations, methods, and selection rules (GOMS; Card, Moran, & Newell, 1983).

One problem with existing methods of TA is that the granularity is too small in the definition of the tasks that are traced (e.g., entering a single command, pressing the right mouse button). We have often observed that a minimal amount of key-pressings and mouse-clicks are performed during a much longer period of professional interaction with the work task, containing information search, judgment and decision making, and so on, in a typical period of time. Larger concatenated tasks make the work context more important. In this way work can be considered to be simple tasks, to be carried out as efficient as possible, and, not to forget, with movement between the tasks as efficient possible. This task switching, noticed in some publications (Card & Henderson, 1987; Henderson & Card, 1986), but seldom emphasized in TA methods, is important, if not crucial for efficient human-computer interaction (HCI) in a specific work setting. These important perspectives on the work activity could be reached by dividing the work into larger units, which we will refer to as *work tasks* (See Appendix A for definition), continuous in time, with a starting point and an endpoint, and terminated by a decision (See Appendix A for definition).

In an article on cognitive engineering (Norman, 1986), the understanding of the fundamental principles behind human action is described. Performance relevant for the development of engineering principles of design and systems that affords pleasurable engagement, are outlined. We have, in relation to this, stressed the difficulties in capturing the user's mental models and such problems related to information system engineering (Gulliksen, Sandblad, & Lind, 1996). The gathering of informal design knowledge, and the problems of its application in information system engineering requires a method to extract directly design controlling knowledge.

The contextual inquiry method for participatory design (Holtzblatt & Jones, 1993) is a step towards the solution of the problem. It supports the production of general purpose systems and provides us with a way to work for short periods of time with users at multiple, geographically dispersed customer sites. Three main principles guide the contextual inquiry process: context, partnership, and focus. The process

provides an understanding of the nature of the user's work through inquiries with users during ongoing work. The principle of partnership recognizes that only through dialogue with the users can designers become aware of their experience of work and use of tools. The choice of focus is also important when inquiring the users. The analysis results in a description of the work and identifies problems concerning the users work and tools. Contextual inquiry is especially useful when designing a new version of an existing software product. Other participatory design methodologies have been defined with a general assumption that users should not only be involved but in control of the analysis and design process (Schuler & Namioka, 1993; Greenbaum & Kyng, 1991). To some extent, this has been taken as far as the introduction and cooperation with the end-users union organizations to amplify the legal and democratic right of the users to influence and control their future work situation, in what is known as *cooperative design* (Bjerknes, Ehn, & Kyng, 1987).

Users are undisputedly important in the user interface analysis, design and evaluation phase. However, according to our observations, users are not always aware of their work behavior, nor are they experts in design and HCI. This is why a user-controlled design methodology sometimes can have negative effects on the usability of the resulting information system. Difficulties related to user participation in the development process can be a severe obstacle to efficient user interface design (Grudin, 1991b). This does not mean that user participation in the development process should be avoided. Rather, user participation should be made more efficient by restricting it to stages where users can participate efficiently. Users are essential for modeling the work activity domain (See Appendix A for definition) and for stating the goals of work and how work is performed today. Users are also necessary in the iterative design cycle for the testing and validation of prototypes and systems. It is, however, our belief that users are not equipped to actually do design. User interface design should be performed by skilled designers in close cooperation with the users in the right phases and with a deep understanding and knowledge about the work domain. By introducing a dedicated user interface designer, unnecessarily poor design solutions, due to user requirements being in computer terminology (e.g., demand of specific icons and windows), could be eliminated. User interface requirements and information needs must be stated in terms of the work activity.

In in-house development projects it is very common that the large amounts of documentation, produced as results of the TA, build a wall between developers and users, particularly when developing according to the waterfall model. It is our experience that the documentation produced is seldom used in the interface design phase, either due to its being insufficient or due to its being too large and extensive and hence impossible to review. It is here that the analysis of information utilization (AIU) documentation can constitute a valid substitute or addendum for our special purpose of controlling the design. AIU produces documentation that is directly

related to the design and hence easy to get an overview of and use in design and development.

Tracing the actual nature of computerized work can be performed by emphasizing the importance of regarding the entire work domain, focusing on all aspects of work: organization, competence, work activity, and information technology. This issue was first raised a long time ago (Leavitt, 1958), but it is so seldom actually considered in real-life development. Therefore, this article describes a method of analyzing information utilization, demands on essential prerequisites for the method, and resulting descriptions from the process.

2. The Need for a Focus on Information Utilization

Every ambitious system development project in an organization performs some kind of TA to be able to identify bottlenecks, problems, and development opportunities. In connection with this TA, a data model is created in which data, used in the organization, and their mutual relations are defined. Our experience from development projects in working life is that with this TA, the analysis process is often regarded as completed and the design and construction phase starts. Many unanswered questions and missing knowledge on the work domain are left to be solved in the design and construction phase. However, not only which data that are used in the HCI but also how these data are used is essential when designing an effective user interface to a specific work task. There are two main reasons for this. First, a discussion that can be summarized as follows:

- Case handling, the main focus of our research (Gulliksen, 1996a), means that the individuals performing it make decisions and judgments.
- Decision making and judgments are cognitively demanding processes that require conscious attention and creativity.
- Human cognitive capacity is limited regarding these processes.
- Human activities vary concerning the amount of cognitive conscious capacity that is allocated (Rasmussen, 1983; Reason, 1987).
- By analyzing how information is being used in case handling, it is possible to create human-computer interfaces that can be handled with a minimum of required cognitive capacity. The interface can present information in a way that can be decoded using a minimum of cognitive capacity, leaving as much attention as possible for work-related judgment and decision making.

Second, an observation that humans by themselves create and customize special tools (e.g., forms, scrapbooks, spreadsheet-documents, card index, etc.) and methods that are adapted to the work tasks in the sense that their environment allows such an adaptation. It is important to understand these adaptations and how they work in order to be able to integrate them into a new system. This does not mean that one should mechanically transfer paper-based features to a computerized form but strive to find functional adaptations that otherwise could not be found.

2.1. Skilled Professionals and Cognitive Workload

The information systems considered here can be characterized as systems aimed at supporting administrative work in a broad sense. In such systems, we include staff and economic administration, systems for businesses, banks, logistics, health care, and so on. We do not consider more general-purpose systems such as word processing, spreadsheets or pure data-entry systems.

Skilled professionals using computer artifacts in the work process can find unnecessary cognitive work load in the interface a severe obstacle. When computerized information systems are used, for example in case handling work, the purpose of the work is never to operate the computer. In our research (Gulliksen, 1996b) we have seen examples of computer systems in which up to 80% of the work time is spent managing the interface. The problem-solving process is constantly interrupted by the need for re-design of the interface, opening, shrinking and moving windows, starting different applications, locating and interpreting information, and so on. This, of course, results in low efficiency and bad user acceptance, a high level of anxiety and stress, and even health problems (Johnson & Johansson, 1991). The computer is a tool that will be used and appreciated only as long as it efficiently supports the purpose of the work (e.g., to perform judgments and decision making on case-related information). Therefore, the interface should be designed on the basis of optimization of work activities instead of just optimizing the use of the computer.

Reading from paper documents allows knowledge to be gained automatically by pattern recognition and by task-related encoding of the information media characteristics (Nygren, Allard, & Lind, In press; Nygren & Henriksson, 1992). When reading from a screen the corresponding knowledge is often gained by effortful cognitive processes. This problem can be avoided by a careful analysis of the reading task into automatic and non-automatic components (Schneider & Shiffrin, 1977), followed by a dedicated user interface design with information presented in a perceivable rather than in a readable way. Design must be based on an analysis based on an attempt to minimize the user's cognitive load.

The practical consequence of this is that such a design must be based on an analysis of how information is being used in the actual work context for which the application (or artifact) is developed. The design must be performed in such a way that the management of the interface can be as automated as possible. It is then possible that the user can make optimal use of his or her creative and problem solving abilities. In this way, the computer system will be "transparent" and the user can concentrate on the work. We say that the interface must be obvious to the user (Nygren, Johnson, Lind, & Sandblad, 1992).

2.2. Work Characteristics, Case Handling and Decision Making

In current computer system development, only a small portion of the total amount of resources is put into user interface improvement (Nielsen, 1993). The outcome of human-computer interface investments has not always proven to meet its anticipation. Huge investments in white-collar business work situations have, in many situations, failed to yield the desired increase in efficiency (Willoch Management Consultants, 1993). A major part of the work performed in the domains we are interested in can be described as making judgments and decisions. Judgment and decision making is a demanding cognitive process (Gulliksen et al., 1996). Because human cognitive capacity is limited, computer systems should support such decision making. The performance of a work task and control of the user interface can be regarded as two concurrent tasks competing for the users' cognitive resources. The main objective when constructing interfaces intended for use in judgment and decision-making situations is to make sure that the interface requires a minimum of cognitive processing and physical actions from the user during this work task. The need for paging or scrolling, as well as calling up additional windows, replying to modal dialogue boxes, resizing and rearranging windows, and so on, should be minimized. Decision-relevant properties of data should be coded so that they demand fewer cognitive resources for interpretation. This can be done by letting the content of a variable control the appearance of it in a systematic and relevant way; that is, by controlling font, size, color, and others.

Central to the creation of efficient user interfaces is minimizing cognitive load imposed on the user when performing work tasks. To do this, we have argued elsewhere (Nygren et al., 1992) that the user's work tasks must be analyzed in terms of which decisions they make. Examples of the kinds of decisions we refer to include rejecting or accepting an application, sending or not sending a form to someone else for consultation, and so on. From the time the work task is started and until a decision is reached, the cognitive resources of the user are heavily strained and the manipulation of the interface should be kept at a minimum.

3. AIU

AIU describes tasks of decisions and judgments that are a part of a certain type of work and how data is used for solving these work tasks (Lind, Sandblad, & Nygren, 1991). In contrast to the MBI-method (Hugosson, Hesselmark, & Grubbström, 1983), which also analyzes a work activity in terms of goals, decisions, and information, the decisions and judgments are here defined per actor (category of work) and not independently of who makes them. Information utilization aspect description and analysis has to be done in the actual work environment. Our approach to this challenge is to involve representatives from the work domain. The analysis of work and the formulation of the resulting descriptions should be performed in close cooperation between user representatives and experts in the methodology. AIU is performed with one or a few persons representing an actor in a work activity.

Some parts of the work performed in an AIU can be neither formalized nor captured in a method. An analyst observes another person's work, and through that acquires some understanding of the purpose and contents of the work. Hereby, he or she acquires knowledge and experience that can be useful when designing the system or the user interface. It is not always obvious which information will be useful knowledge for design when doing this analysis. That is why we strongly recommend that the person performing the AIU is the same person that is responsible for, and participates in, the design of the user interface. The description of AIU should be based on what is performed, how it is performed, and how the results are documented.

3.1. Prerequisites for AIU

A basis for AIU is a TA of a more traditional nature (Johnson, 1992). Here work activities are described and analyzed with respect to their contents, work organization, and existing problems of a different nature. An existing application model (an implemented, functioning, object-oriented data model containing data and methods to specify the behavior) and an organizational model can be assumed. The data model puts limits on the design space. If AIU requests changes, the design of the user interface can be enhanced, assuming that the data model is specified in a modifiable way. Here, we are going to specify methods for AIU and a taxonomy for specification of the work model, which is the formal outcome of AIU.

3.2. Methodological Steps of AIU

A detailed AIU of work activities in the defined domain is performed. There are a number of problems involved in analyzing how professionals utilize information while performing work tasks. Because the reason for performing such an analysis is

to reduce the cognitive workload, the work tasks must be defined in relation to individuals and their behavior. What the designer want to establish as a prerequisite to design is major decision-making situations of the intended users in relation to the task proper. Unless the future system incorporates modules performing such decisions automatically, which is fairly uncommon, these judgment and decision-making processes will be present in the future system also, although they will be supported by other tools. This relates to work situations in an hierarchical fashion. On one level, the granularity is high, suitable for construction of interface elements. A level of low granularity is suitable for the construction of complete interfaces based on these elements. AIU is performed by doing the following:

1. First find a representative sample of individuals who are skilled at the work for which a new computer system is being developed.

2. Make a general description of the contents of the work seen from the perspective of the actor.

3. Describe and collect copies of all sorts of information carriers that are used today, for example, forms, notebooks, screen prints, and others.

4. Describe all existing routines for handling the information carriers used today, for example, portable archives, white-boards, mailboxes, data routines, files and folders, mail delivery service, and so on.

5. Describe the judgment and decision-making routines that are involved in work. The work tasks the users perform while working, must be distinguished and categorized.

6. Describe information sets (variables) that are used in each work task and how these relate to the data model. These variables should all be simultaneously presented by the interface when a certain work task is performed.

7. Describe how the variables described above are used today for each work task.

8. Analyze the material from each work task in terms of:
   - Demands on the simultaneousness of data.
   - Value range and features of each variable.
   - What must be performed, for example, how decisions are documented.

   This information helps the interface designer to choose representations for the containers of the variables.

9. Analyze the material in terms of the work tasks that have to be performed simultaneously. This defines work situations.

The AIU must be sufficiently complete to cover all relevant work activities in the domain. We do not only describe what information is being used by whom and when, but also important aspects of how it is handled by the professionals using it in different contexts. This analysis forms the basis for user interface design decisions. Based on this analysis, a set of interface components covering the field of possible applications is defined and designed. The design is made so that the components can be identified and used with a minimum of cognitive load. The specification of basic and more complex interface components is a step-by-step procedure.

To construct such a design, the work model must include (a) a list of work tasks performed, as defined by decisions that are made and which work tasks that may have to be performed simultaneously; (b) a list of variables, in terms of the data model, that could possibly be used in performing each work task, their priority, and features; (c) a list of actions needed to manifest each decision; (d) a list of naturally occurring work-situations related to different actors, defining sets of work tasks that usually (or possibly) are performed concurrently; and (e) a scenario describing the entire work situation over a typical day. This work model can then be used by the interface designer in the creation of suitable interface elements, such as screen documents or forms, or larger entities such as work-spaces or rooms (Card & Henderson, 1987). The dialogue needed to use these documents and workspaces to perform the tasks is also essential in the design.

4. How Is an AIU Performed?

The foregoing statements are mostly self-explanatory and simple to perform; others, however, are more complex. Next, we describe some of the more complicated concerns and a strategy that we have found useful.

4.1. Understanding Individuals Work and Their Work Tasks

The keywords in AIU are *humility* and *respect*. All work, unless it has been mutilated through an unsuccessfully computerization, contains possibilities for entering deeper. The user acquires an increasing amount of skill in handling and mastering the complexities that are the essence of work. This means that one cannot perform an AIU purely through observation, and one cannot completely understand a person's work only through AIU.

Too often, one finds persons that regard themselves as experts on how a category of professionals perform their work and what they do, without ever having performed the work themselves. These convictions have always proven to be erroneous. This means, among other things, that the only way to perform an AIU is to cooperate with the actual category of professionals. Preferably, these persons should have a great

interest in their work and being interested in developing it further. Computer experiences are, in most cases, totally irrelevant.

This means that the person responsible for the AIU has a main task of listening and trying to understand and document everything that occurs during a work session. The individuals analyzed must be lead into all side tracks and exceptions that exist in work. Informally speaking, people seem to spontaneously describe routine regularities in their work. Of course, this regularity is important, but usually it describes a minimum of the actual work time. The rest of the time is filled with "exceptions" and "special cases". Capturing these parts of the work is of utmost importance and often neglected in more traditional work descriptions. The lack of flexibility in user interfaces and computer systems that can be the consequence of such an oversight has been noted to be a common reason for dissatisfaction with computer systems in working life.

Another aspect of AIU is that difficulties occur due to people's abilities to automatize parts of their behavior. What we see as a possibility for making work more efficient and decreasing cognitive work environment problems is something that aggravates AIU. This is strongly related to what we call *tacit knowledge*, knowledge that is not easily communicated to other individuals due to its complex structure (Polanyi, 1958). AIU does not capture these automatized cognitive processes but maps the assumptions for them. This is performed by studying tools that are used in work and by observing the person performing the work and then making the person aware of performances that can be related to automatic processing. The person can then typically tell the purpose of the operation and further dialogue can reveal at least some of the prerequisites for the automatic processing to be performed.

4.2. Identifying Judgment and Decision-Making Situations (Work Tasks)

One of the fundamentals of AIU is to identify judgment and decision-making situations. Unfortunately, we do not have a distinct method for this. An important cue is when a decision is performed and when the actor regards that one stage in the handling of a case has been fully achieved and that the case can be postponed, if needed.

We find work tasks by noting whenever users document something. Every time we change the information contents in a case (i.e., not just pure memory relieving or copying) should be regarded as a decision, the end of a judgment. It could be scribbling, resorting information, writing, clicking, or some other action. The beginning of such a work task is then established by interview techniques.

4.3. Describing the Variables That Are Used

By describing the tools the professionals have at their disposal and discussing the work tasks in relation to these, it is possible to decide on a number of information entities (variables) that are essential for every decision. Users tend to describe general properties of their work, leaving out exceptions. Because of this, the data collection might be very demanding. These variables are given a name and identified based on how they are used or what they are called in the work context. In the description of the analysis results, these variables must be related to the data model for the organization.

Important variables that are used, perhaps without the users' own awareness, need to be documented in the analysis even if they are not documented in the work situation in terms of a field on a form or even of letters or numbers written somewhere. The way of writing can code a variable. For example, different grades of "reliability" have been efficiently coded in an intensive care work situation by writing less reliable values with a pencil and more reliable values with a pen (Pettersson, 1989). Other variables in a paper-based world can be coded with expression, for example, the author's signature or when it was written. Categorizing these variables demands a lot of attention but is, nevertheless, important because neglecting these variables in the design of the computer support can easily aggravate user performance or decrease efficiency.

In computerized work, where a lot of information has been removed due to the simplicity of older data models, it can be useful to question whether certain variables that now are not present could be important and exemplify variables of this type. This type of question can bring forward earlier known but unexpressed information needs.

The list of variables that is put together should not attempt to describe how the variables are used for decision making. Further, all variables should be included even if they are only used a few times. The result should be the total amount of variables from which the professional selects a subset, on which the decision is based. Every identified variable must be classified considering features it can have. The following scheme can be used for classification (Foley, 1990).

1. Continuous variables.

2. Discrete variables with an ordered set of values (ordinal scale).

3. Discrete variables with an unordered set of values (nominal scale) This category must be distinguished based on whether the values can be mutually excluding or not.

The set of values must also be described. In cases where a variable can be assigned a limited number of values (e.g., less then 20) each of these must be described. Important features of the variables might not be captured in the object data model, for example, such features that are specific to the group of users in question. An example is the property of having status of an estimated but not validated value, estimated and validated value, objectively measured value, and so on. This can be used for defining the form of presentation as well as for the refinement of the data model.

4.4. Defining How Each Variable Is Used for Each Work Task

The relative frequency of the use of the variables can be specified by interview and observation. This can be hard because, for the same reasons as before, people well acquainted with their work tend to automatize parts of their behavior in a way that makes it impossible for them to give a good description of what they do. It is also difficult for them to mention exceptions, as mentioned earlier. A good method is to observe and ask questions while (or after) the work is performed. This can also help the person performing the work to focus on what he or she does.

4.5. Describing How Information Is Used

AIU does not aim at describing decision rules that are used in judgments and decision making but at describing how the information carriers (computer screens, paper documents, etc.) are used, handled, and organized concretely by the users in each work task. For example, two information carriers are placed beside one another, but the third is brought forward only if needed, or, a person holds his or her fingers on certain pages in preparation to turn them rapidly. If a computerized work situation is referred, this will be a description of, for example, how professionals move between screens to be able to make judgments, what causes interruption or irritation in the line of thought, which information carriers of a non computerized nature are used simultaneously with the computerized material, and how these are used.

4.6. Analyzing the Entire Material Per Work Task

As mentioned before, one should try to make a conciliation of the information in the descriptions made in terms of three different things.

Demands on simultaneous presentation of data.

Simultaneous presentation of data needed for a decision is a great aid for the actor (Lind, 1991). Based on the collected information, every judgment and decision-making situation and every variable belonging to it needs to be worked through

together with the domain expert. This discussion results in a classification of each variable in two different levels. Level 1 contains all variables needed frequently or rather frequently for the decision, and Level 2 contains variables that are used infrequently. These levels will then contain a summarized description of the demands on simultaneousness. The variables that end up in Level 2 should not be regarded as variables that can automatically be hidden in the new user interface to be revealed on demand. On the contrary, a basic rule should be that all variables must be shown. A classification into Level 2 should be considered only as a possibility for compromise in the design phase if other demands on screen space become too big. The classification into Level 2 should be made with considerable caution if there is a risk that a variable can have a substantial signal value even if this is seldom seen.

Value set and features with the included variables.

For the descriptions just described to be useful for construction of user interfaces, each variable must be related to the data model; that is, the variable needs to be described in terms of the information that will be stored in the system. One will eventually find some variables or features that are not present in the data model. These facts must be analyzed together with the people responsible for the data model to see to it that the data model is complemented. The values for each variable must be discussed together with domain experts to be able to tell which values are important to show and code. Visual cues are needed for showing different types of information, just as for screen space. Compromises are often needed and then it is useful to analyze the degree of necessity for specific demands (e.g., showing values of the variables).

What needs be done for every judgment and decision-making situation.

The possibilities for action that are a part of a decision-making situation need to be supplied with adequate appearances (e.g., pointing, button, key-sequence, part of pull-down menu, etc.) in the user interface design process. For this, the designer needs a list of all the actions that need to be performed in every decision making situation.

4.7. Demands on Simultaneousness Between Work Tasks

Often there are work tasks that logically are unrelated but that are needed to be performed simultaneously due to external conditions such as organizational structure, telephone hours, and so on. Such conditions are relatively easy to specify but very important for the interface design process.

4.8. Analyzing Future Work Situations

Most of what has been described above is valid in cases when the development project concerns an existing work situation as well as a future one. When the AIU concerns a future work situation, for example, the design of a support system for a completely new service in an organization, of course no work situation exist. In such cases it is sometimes possible to base the AIU on such parts of existing work situations that correspond to the new situations. Other aspects of the new work situation can be specified in user-centered modeling sessions. Here the cooperation with experienced users is essential. The use of different types of scenario-based techniques (Carroll, 1995) can be a solution. Detailed scenarios containing the new work situations can be used as a base for experienced users to define how information-handling routines should be performed under the new circumstances.

5. Formalized Development Models for Describing Work Characteristics

The documents that need to be manufactured are listed relating to different activities and formal specifications (Figure 1). Following, the different models used are briefly described for the understanding of the context of the AIU.

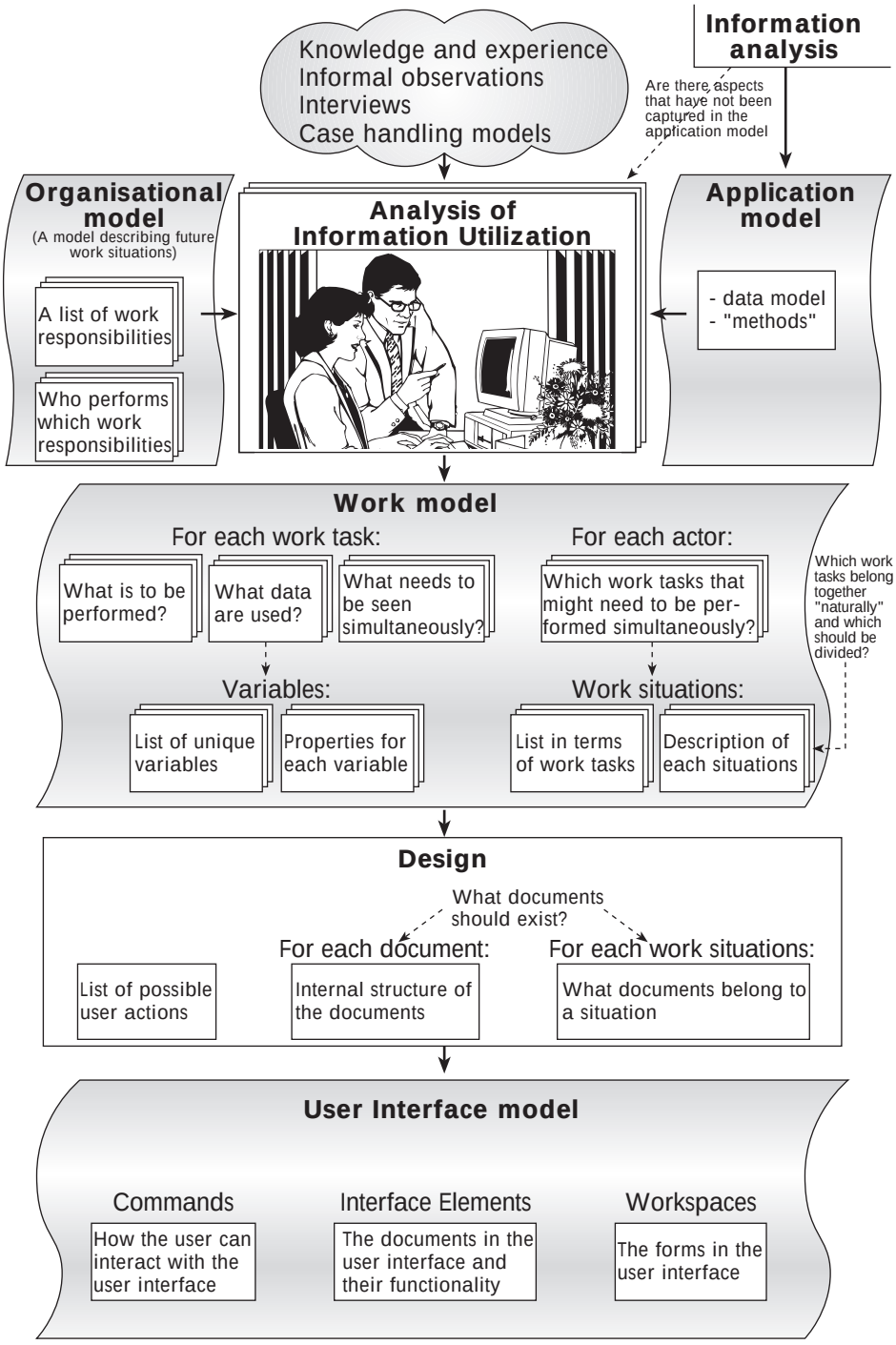

Figure 1. How AIU is documented (work model) and how the results are used in the design process.

5.1. The Organization Model (OM)

An OM has to exist in some form before development work starts. How this model has been established and by which methods it has been developed are not interesting at this point. It is, however, important that the model describes the future work situation if it is to be effective in development work. An OM is often specific for a work activity, depending on the fact that the organizational structure in rather similar work activities has shown to be very different. It depends on there being many ways to construct and implement OM. Many things in the OM are not interesting for user interface design, such as overall distribution of work responsibilities. This is why we have chosen to specify requirements on the OM (Gulliksen, 1996a) that is interesting for our purpose; that is, we chose to design efficient user interfaces by performing AIU.

The organization model should at least contain the following if it is to be useful for interface design:

1. The *overarching goal* of the work activity.

2. The *range of users*; that is, what categories of users exist within the organization and what their competence profiles are.

3. Overruling classes of *work situations* in terms of work responsibilities; that is, what is to be performed. The *purpose* of the work is central.

4. Based on these classes, the *category of users* are supposed to perform the classes of work situations should be defined.

Through this one can specify who performs a particular major work responsibility, which is a necessary prerequisite when performing AIU.

5.2. The Application Model (AM)

The AM is, in the traditional sense, a data model, although now extended with some additional methods (e.g., a bank client carries methods to show amount on his account, methods to calculate actual interest rate, etc.) that otherwise should belong to the data base, but seldom do.

It is important to base the information analysis on object-oriented design methods (Booch, 1990; Rumbaugh, Blaha, Premerlani, Eddy, & Lorensen, 1991). One such object-oriented structure of the information system and the user interface is the PAC-model (Coutaz, 1987), which distinguishes three different modules: presentation, abstraction and control (PAC). The presentation module displays the user interface and receives the manipulation interaction from the physical user, and

transforms it into a new concrete appearance of the user interface (e.g., feedback). The abstraction module is the physical appearance of the data model; the database and the functionality of the application. The control module handles the communication and supports consistency between presentation and abstraction. Every presentation component is composed of a set of PAC components recursively. As object orientation increases, the functionality of a computer system is subsequently divided between the user interface and the database. Objects in the database carry their own methods, and the user interface handles all presentation issues, why separate functionality specifications become unnecessary.

With an AM that extends the abstraction module from PAC with the methods just mentioned we can derive a more independent set of interface objects, immediately related to information sets from the work activity and objects in the data model. The presentation becomes "views" of these information entities and the control is transformed into a general communication module between the database and the presentation views. This is analogous to the transformed model of user interface and information system development (Boehm, 1988). The remaining part of the control becomes an interface specification tool that, more or less automatically interprets an AM alongside a description file of the presentation of the user interface into a functioning application and database (Olsson, Göransson, Borälv, & Sandblad, 1993).

5.3. The Work Model (WM)

The WM is the formal physical outcome of the AIU. The following documents need to be manufactured (their contents can be read from the foregoing):

- A list of work tasks identifying judgment and decision-making situations and which work tasks that may have to be performed simultaneously.

- For every work task (judgment and decision) a list of variables that could possibly be used with features and priorities.

- For every work task what needs to be performed in terms of possible user actions (e.g., in terms of information search, data entry, or data search).

- A list of work situations consisting of work tasks relating to the different actors. The work situations are defined based on informal observations and interviews, knowledge and experience and existing work model that needs be followed, as an understanding of the work tasks that "naturally" belong together and those that should be divided.

- A description of the sequence of work seen over a typical day, week, or whatever scope of time is appropriate; that is, not per case (e.g., a scenario).

It is important to notice that no design decisions have been made yet. The specification of the work model is a formalization of observations that was made about how the work was performed today, and it should not contain any design restrictions. Design is a creative process that should be performed without the risk of disturbing valuable domain knowledge.

5.4. The User Interface Model

The interface model, that is, the result of the design process, can be divided into three redundant parts:

1. Interface elements, consisting of the documents in the user interface and the functionality that is incorporated in the documents.

2. The functionality, describing how the user interface is physically manipulated by the user.

3. Workspaces, consisting of the forms in the user interface.

It has been noted that the user faces several different problems when attempting to accomplish several different tasks in a single session (Bannon, Cypher, Greenspan, & Monty, 1983). The concept of workspaces for reducing the mental workload is introduced. This supports the user's activity coordination. This is analogues to workspaces that attempt to show all information relevant for a case handling session simultaneously.

6. User Participation and the Design Competence

User participation in system development is essential (Ehn, 1988; Bjerknes et al., 1987), but for several reasons known to be difficult (Grudin, 1991b). It often can be difficult to identify appropriate users and to receive their full attention (isolating developers from users tends to receive high priority). The developer needs to be motivated to keep contact with the user. To get users to set aside their usual work activities may cause problems; they may never get to use the product; they tend to believe that their influence is very limited; and, they may feel that there is a risk of being rationalized. Where in the development process users can have influence might sometimes be hard to identify. Receiving proper feedback from the users can also be a problem. Above all, with the development tending to be routine, user

participation might be hard to adopt, due to the heavy time restrictions in the development projects.

There is a gap between developers and potential end users that needs to be bridged (Grudin, 1991a), in structured design based on a top-down methodology. A system analyst or human factors expert, with knowledge from system development, the user, and the work activity, could constitute the principal link between developers and users (Grudin, Ehrlich, & Shriner, 1987). Design is often argued to be a matter of craft and experience (Wallace & Anderson, 1993). Most development projects are extensive in time; developers seldom have the opportunity to participate in several development projects to acquire afforded experience.

We have introduced a specialized design competence, to function as a mediator between user groups and development teams in large corporation development projects. It is important that this design expert can participate in every stage of the development process for the best possible conservation of the knowledge acquired in the analysis and modeling phase. Apart from good knowledge of software engineering and human factors analysis methods, the designer must be experienced in HCI and related topics (cognitive psychology, art, graphical design, etc.) Above all, the designer must be able to acquire knowledge of the work domain. A designer of this type can have the total view on design in a large organization and thereby support consistency over applications in a domain.

Based on AIU it is possible to specify interaction requirements and design a user interface that allows the user to use cognitive resources for decision making instead of manipulating the interface. The results of AIU has to be translated into formal design specifications, but there is also an informal part (in natural language) that is impossible to formally specify. These informal results from AIU has to be treated and coded in the design process. User interface design is a question of assembling work decisions with the information needed for each decision, adding necessary special features. It is therefore partly an issue of art and taste and should never be performed automatically. When formalizing and communicating the results of the different analyses, important information might be lost (Gulliksen, Lind, Lif, Sandblad, 1995). We need to enhance the communicating and interpreting aids for transfer of knowledge between the different areas that have been subject to analysis. This communication is, to some extent, a question of design. We have defined the concept of design in HCI as the creation of a formal description (e. g., program code, formal language) of appearance and functionality based on partly informal (to a large extent) results of analysis (Gulliksen et al., 1995). The design of the work model will then be the process of formalizing the outcome of the AIU. As information about the actual work might be lost when designing formalized descriptions, communication gaps in design occur. Methodologies for design can bridge or narrow these gaps, improve knowledge communication, make development more

efficient and economical, prevent unnecessary work, and enhance the production of usable interfaces.

With formalized viable methods for AIU, a framework for the specification of the work model, is defined. The domain specific design methodology (Gulliksen & Sandblad, 1995) can enhance the interpretation of the work model. But, to preserve a maximal amount of the informal knowledge of the work, which the analyst might not be aware of, we recommend that the person conducting AIU be the same person that is responsible for the design.

7. An Example

To illustrate how AIU can be performed, we give an example from a research project that aims at designing illustrative prototypes of real-life work for the Swedish National Tax Board. The types of work that we here consider involve case handling resulting in decisions on personal income tax declarations. The prototype designed did not aim to describe a real work situation but to serve as an example to illustrate the benefits of conducting AIU to capture design-controlling aspects of the user interface. The example does not fully cover a work situation, and it only presents examples of the contents in each list included in the work model.

7.1. Example of a WM Resulting From AIU.

A list of the total amount of work tasks in terms of judgment and decision-making situations is produced. The list should include every work task that may occur while performing the work (Table 1). It is important to identify which work tasks that may interfere with other work tasks (e.g., a telephone call from one client may interrupt the user while he or she is producing a tax decision concerning another client). In table 1, work tasks with the same number may have to be performed simultaneously.

Work Task	Simultaneous
Producing an inquiry	1
Message of proposed action	2
Producing a tax decision	3, 4
Giving service	4
Reference information search	3, 4
Electronic mail	1, 2, 3, 4, 5, 6
Organizing the staff	5
Register letters	6
…	

Table 1. A List of Work Tasks in Terms of Judgment and Decision-Making Situations and the Demands on the Simultaneousness of the Work Tasks Performed by Different Actors.

A list of the total amount of variables used while performing each work task is documented in terms of the data model (Table 2). Features that are not encountered in the data model is also included in the list. The structure of this is heavily depending on the nature of the data model.

The priorities of each variable are also documented. Variables with priority = 1 has to be visible on the screen all the time. Variables with priority = 2 may be hidden if there is not enough space on the screen.

Variables	Priority	Features
Personal ID	1	
Name	1	
Address	1	
Income tax declaration	1	Signed, notes
Total income	1	Over 300, 000 SEK/year
Joint taxation	2	
…		

Table 2. A List of Variables, With Features and Priorities, for the Work Task of Producing a Tax Decision.

Possible user actions that needs to be performed for each work task are extracted in the next list (Table 3).

Possible user actions
Enter personal identification number
Calculate new interest rate
Activate a reminder
Tax decision
…

Table 3. Possible User Actions for the Work Task of Producing a Tax Decision.

By using the lists above it is possible to identify the different work situations handled by each actor. Work tasks that may have to be performed simultaneously (Table 1) should be included in the same work situation. The work situations, the corresponding work tasks, and actors are documented in table 4.

Work situation	Actors	Work task
Case handling	Work activity expert	Producing an inquiry
		Message of proposed action
		Producing a tax decision
Evaluation	Work leader	Organizing the staff
	Case administrator	Evaluate the work
…		

Table 4. A List of Work Situations and Corresponding Work Tasks With the Possibility of Relating It to Different Actors.

Finally, the scenarios are illustrated with descriptive free text describing the sequence of work in a typical work day; they illuminate aspects of task switching and other, possibly overlooked, factors that can affect cognitive load (Figure 2). Based on these tables and the scenarios, illustrative mock-up prototypes can be used to validate the previously mentioned lists by giving them certain design features. It is important to note that no design decisions have been made while producing these lists.

Scenario

A user arrives at work 8 p.m. Usually approximately 20 cases are waiting to be handled. All material concerning one case is collected in a folder. A quick look at each folder is made to decide whether any urgent cases need to be handled. Usually one or to cases are urgent and these are the first to be handled. The first urgent case is here to produce a tax decision. Therefore the information in the income declaration is studied. First, the user checks to determine if there is any missing information. If so, a letter is sent to the customer requesting complementary information within 2 weeks. Then the user can verify the calculated taxes. After that, the case handler judges the reliability of the data by comparing it with information on the customer found in the database. A decision is then made whether to approve the tax declaration or not.

When the urgent cases have been handled the user checks the mailbox,...

Figure 2. An example of a scenario describing a typical work situation for a user.

7.2. Design Based on the WM From AIU

Based on the various lists described above, the design process could result in the prototype shown in Figure 3.

In this prototype each actor has access to a specific subset of workspaces for the possibility of performing the relevant tasks for that category of work. The icon buttons on the top (1) are routes to the different workspaces. Each workspace correspond to a work situation documented in Table 4. The work situation in this example is case handling. In each workspace the user has to reach all information and all devices needed to solve the different work tasks. In this example the user is producing a tax decision. The information that has to be visible while performing that work task are defined in Table 2. In the same table the priority of the variables are documented. Information with priority = 1 is visible on the screen at all times. There are also some examples of information with priority = 2 in the interface. This information is not visible on the screen but can be reached via a link or a button. In this example the information about joint taxation can be reached via a link (2). The features of the variables described in Table 2 are also visualized in the interface. One example is the different modes of a case (3) shown as different styles of the font (bold, italic, underlined). The simultaneousness demands from Table 1 is solved as shortcuts to exit every work session. The status picture icons (4), to the right of the

personal identification element, are designed to be a reminder and a way back to the task from which case handling was interrupted. Table 3 is the basis for the construction of the dynamic functionality of the user interface, which is a property that cannot be illustrated by a paper prototype. The scenarios described in free text is of great help when making the different decisions on how to design the prototype.

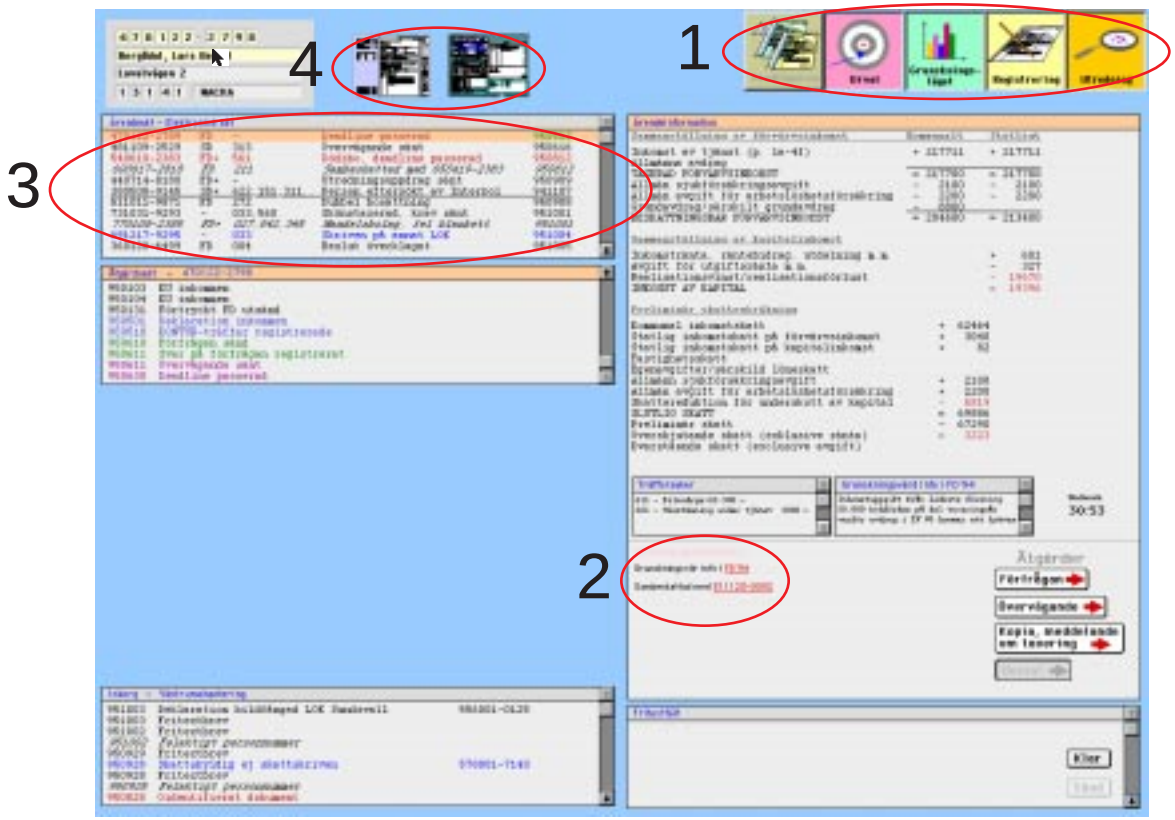

Figure 3. An information system prototype designed based on a careful AIU. The circled interface elements each represent design solutions based on the resulting tables from AIU. This prototype illustrates an attempt to minimize the cognitive load imposed by the interface, and it was positively judged by the user representatives. (Colour plate 1).

8. Evaluation

It is important to evaluate AIU with respect to reliability and validity of the method. Reliability is related to observer agreement; that is, if similar results are obtained by different observers of the same users and work situations. Validity is related to whether or not the method really measures the variables it is intended to measure.

The reliability, or the observer agreement, has been tested in connection with some application development projects. Because the method allows observers to perform analysis in different ways, and because of the impossibility of observing the same users more that once, the evaluation becomes complicated. However, the resulting documentation from AIU can be compared and evaluated. Preliminary tests show that the observer agreement is acceptable, at least for experienced observers. The validity is a measure of if AIU really fulfills the purpose, that is, to contribute to better and more efficient design of user interfaces. This can indirectly be evaluated. We have previously shown the importance of reducing the cognitive load induced by the interface (Lind, 1991; Nygren et al., In press). If AIU can contribute to interfaces that require less cognitive efforts to understand and handle, the purpose of the method has been fulfilled. This has been tested in application development projects, and results show that the criteria of minimizing cognitive load can be reached if the design process is based on findings from an AIU.

A meaningful and strictly scientific evaluation of a method of this nature, is difficult to perform. It would, for example, require two independent development projects to be completed in parallel, with an evaluation of the resulting interfaces with respect to long-term effects and user efficiency. Such evaluation procedures are not realistic, for methodological as well as of economic reasons. AIU, however, has been judged to be a very useful tool and will be incorporated in system development methodologies at, for example, the Swedish National Tax Board.

9. Discussion

The main purposes of AIU are to capture aspects of how information is being used and certain features of the information that has not been identified in the data modeling process and to constitute the basis for better user interface design. Informal aspects and experiences encountered in the AIU are important for user interface design. In the concept user interface design we also include parts of the description of the functionality. In this way the AIU provides complementary information to the TA, just as to the data model. AIU is important because it makes it possible for a number of domain experts (end users) to analyze their work and view it from the outside. In the different work situations we have studied with AIU,

several suggestions for improvement have been directly detected and effectuated. This is, of course, not a main purpose of the method, but a valuable side effect.

So far, AIU has only been practiced on a relatively small scale. However, it has been very useful both as a complement to existing design methodologies and as an independent method for capturing domain knowledge for user interface design. Because the method deals with establishing a constructive dialogue with the observed person, and to understanding complex knowledge and work procedures, there is no use in describing the method as a "cookbook" approach. It is not possible to describe how to conduct the observation interviews in detail; we can only specify which aspects that should be considered and give strict guidelines for how to document the results. It is up to the individual to acquire appropriate knowledge and develop an individual pattern for the interviews. Even among the authors of this article the procedures for conducting these observation interviews vary. All performed AIU differently and the media for recording the results vary (e.g., written documentation, video, tape recording, etc.). There are a number of important factors of the interview technique that affect the response from the interviewed participants, such as giving and receiving confidence, but the resulting work models are much alike, due to their structure being so specified.

AIU is meant to be incorporated into a more complete design and development model (Gulliksen et al, 1995; See Figure 4). The result of the analysis (the WM) should, together with the OM and the AM, constitute the basis for design of a user interface. It is important that the design process is an iterative process involving AIU, establishment of a first prototype and domain specific evaluation of the prototype (Lif & Sandblad, 1996; Åborg, Sandblad, & Lif 1996). In this way, by an iterative process of analysis, design, evaluation and user testing, the possibility of creating a usable and efficient interface is increased.

AIU is also an important step in the domain-specific design methodology. Domain-specific design (Gulliksen & Sandblad, 1995, 1996; Gulliksen, Sandblad, Johnson, Lind, & Nygren, 1993; Olsson et al., 1993) is a method for increasing the efficiency in the development of end user applications in a specific work domain by establishing a domain-specific style guide. AIU plays an important role in the development of the style guide by applying the method in a somewhat revised form. The advantage of a domain-specific style guide, when developing several different applications in an organization, is obvious (Gulliksen & Sandblad, 1995). The development times decrease, leading to reduced development costs; the interface consistency and the possibilities for efficient user participation increase as a consequence of a domain-related terminology defined by the style guide. Only a limited AIU has to be performed, based on the existing elements of the style guide, when developing a new end user application.

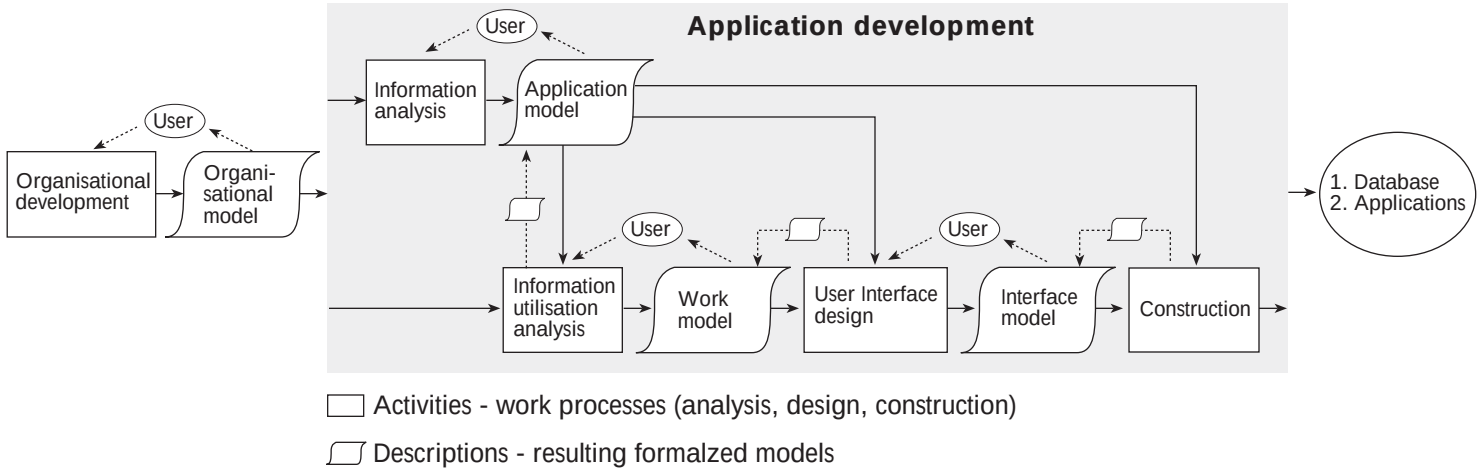

Figure 4. Iterative experimental system development, including consecutive activities and the descriptions they produce. Feedback and possibilities for participatory design are also illustrated.

References

ARKES, H.R., & HAMMOND, K.R. (EDS.). (1986). *Judgement and decision making: An interdiciplinary reader.* Cambridge, England: Cambridge University Press.

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

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

BOEHM, B.W. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer, 21* (5), 61-72.

BOOCH, G. (1990). *Object-Oriented Design with Applications.* Redwood City, CA: Benjamin Cummings.

CARD, S.K., & HENDERSON, A. (1987). A Multiple Virtual-Workspace Interface to Support User Task Switching. In *Proceedings of CHI' 87 Conference on Human Factors in Computing Systems* (pp. 53-59). New York: ACM/SIGCHI.

CARD, S.K., MORAN, T.P., & NEWELL, A. (1983). *The Psychology of Human-Computer Interaction.* Hove, England: Lawrence Erlbaum Associates, Inc.

CARROLL, J.M. (Ed.). (1995). *Scenario-Based Design: Envisioning Work and Technology in System Development.* New York: John Wiley & Sons, Inc.

COUTAZ, J. (1987). PAC – An Object-Oriented Model for Dialogue Design. In H.J. Bullinger, & B. Schackel (Eds.), *Human-Computer Interaction, Proceedings of the 2nd IFIP Conference on Human-Computer Interaction INTERACT' 87* (pp. 431-436). Amsterdam: Elsevier Science Publishers.

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

FOLEY, J.D. (1990). Dialogue Design. In J.D. Foley, A. van Dam, S.K., Feiner, & J.F. Hughes (Eds.), *Computer Graphics: Principles and Practise* (pp. 391-431). Reading, MA: Addison-Wesley.

GOULD, J.D., & LEWIS, C. (1985). Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM, 28* (3), 300-311.

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

GRUDIN, J. (1991a). Interactive Systems: Bridging the Gaps Between Developers and Users. *IEEE Computer, 24* (4), 59-69.

GRUDIN, J. (1991b). Obstacles to User Involvement in Software Product Development, with Implications for CSCW. *International Journal of Man-Machine Studies, 34,* 435-452.

GRUDIN, J., EHRLICH, S.F., & SHRINER, R. (1987). Positioning Human Factors in the User Interface Development Chain. In *Proceedings of CHI+GI 1987 Conference on Human Factors in Computing Systems* (pp. 125-131). New York: ACM.

GULLIKSEN, J. (1996a). Case Handling Models as a Basis for Information System Design. In C.A. Ntuen, & E.H. Park (Eds.) *Human Interaction with Complex Systems-II* (pp. 225-256). Norwell, MA: Kluwer Academic Publishers.

GULLIKSEN, J. (1996b). *Designing for Usability - Domain Specific Human-Computer Interfaces in Working Life.* Unpublished doctoral disertation. Uppsala, Sweden: Uppsala University.

GULLIKSEN, J., JOHNSON, M., LIND, M., NYGREN, E., & SANDBLAD, B. (1993). The Need for New Application Specific Interface Elements. In G. Salvendy, & M.J. Smith (Eds.), *Proceedings of the 5th International Conference on Human-Computer Interaction, HCI International' 93* (pp. 15-20). New York: Elsevier Science B.V.

GULLIKSEN, J., LIND, M., LIF, M., & SANDBLAD, B. (1995). Efficient Development of Organizations and Information Technology – A Design Approach. In Y. Anzai, & K. Ogawa (Eds.), *Symbiosis of Human and Artifact. Proceedings of the 6th International Conference on Human-Computer Interaction, HCI International' 95* (pp. 951-956). Amsterdam: Elsevier Science B.V.

GULLIKSEN, J., & SANDBLAD, B. (1995). Domain-specific Design of User Interfaces. *International Journal of Human-Computer Interaction, 7* (1), 135-151.

GULLIKSEN, J., & SANDBLAD, B. (1996). Domain Specific Design of User Interfaces – Case Handling and Data Entry Problems. In D. Benyon, & P. Palanque (Eds.), *Critical Issues in User Interface System Engineering* (pp. 21-36). New York: Springer Verlag.

GULLIKSEN, J., SANDBLAD, B., & LIND, M. (1996). The Nature of User Interface Design – The Role of Domain Knowledge. 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* (pp. 167-180). London: Design Chapman-Hall.

HENDERSON, 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, 5* (3), 211-243.

HOLTZBLATT, K., & JONES, S. (1993). Contextual Inquiry: A Participatory Technique for System Design. In D. Schuler, & A. Namioka (Eds.), *Participatory Design: Principles and Practices* (pp. 177-210). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

HUGOSSON, M.Å., HESSELMARK, O., & GRUBBSTRÖM, A. (1983). *MBI-metoden. En Metod för Verksamhetsanalys.* [The MBI Method. A Method for Task Analysis]. Lund, Sweden: Studentlitteratur.

ISO/DIS 9241-11 (Draft). Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 11: Guidance on Usability (1995). Geneva, Switzerland: International Organization for Standardization.

JACOBSON, I., CHRISTERSON, M., JONSSON, P., & ÖVERGAARD, G. (1992). *Object-Oriented Software Engineering. A Use Case Driven Approach.* Wokingham, England: Addison-Wesley Publishing Company.

JOHNSON, J.V., & JOHANSSON, G. (EDS.). (1991). *The Psychosocial Work Environment: Work Organization, Democratization and Health.* Amityville, New York: Baywood Publishing Company, Inc.

JOHNSON, P. (1992). *Human-Computer Interaction, Psychology, Task Analysis and Software Engineering.* London: McGraw-Hill Book Company.

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

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* (pp. 181-193). London: Chapman-Hall.

LIND, M. (1991). *Effects of Sequential and Simultaneous Presentations of Information* (Rep. No. 19, CMD). Uppsala, Sweden: Uppsala University.

LIND, M., SANDBLAD, B., & NYGREN, E. (1991). *Informationsanvändningsanalys* [Analysis of Information Utilization] (Rep. No. 15, CMD). Uppsala, Sweden: Uppsala University.

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

NORMAN, D.A. (1986). Cognitive Engineering. In D.A. Norman, & S.W. Draper (Eds.), *User Centered System Design* (pp. 31-61). Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.

NYGREN, E., ALLARD, A., & LIND, M. (In press). Skilled Users' Interpretation of Visual Displays. *Human Computer Interaction.*

NYGREN, E., & HENRIKSSON, P. (1992). Reading the Medical Record 1. Analysis of Physicians Ways of Reading the Medical Record. *Computer Methods and Programs in Biomedicine, 39*, 1-12.

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* (pp. 235-239). New York: ACM.

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* (pp. 133-139). Washington, DC: Open Software Foundation.

PETTERSSON, E. (1989). Automatic Information Processing in Document Reading. A study of Information Handling in two Intensive Care units. In J. Bowers, & S. Benford (Eds.), *Proceedings of the First European Conference on Computer Supported Cooperative Work, EC-CSCW' 89* (pp. 63-72). Slough, England: Computer Science House.

POLANYI, M. (1958). *Personal Knowledge.* Chicago: University of Chicago Press.

PREECE, J., ROGERS, Y., SHARP, H., BENYON, D., HOLLAND, S., & CAREY, T. (1994). *Human-Computer Interaction.* Wokingham, England: Addison-Wesley Publishing Company.

RASMUSSEN, J. (1983). Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man, and Cybernetics, 13* (3), 257-266.

REASON, J.T. (1987). Generic Error-Modeling System (GEMS): A Cognitive Framework for Locating Common Human Error Forms. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New Technology and Human Error* (pp. 63-83). Chichester, England: Wiley.

RUMBAUGH, J., BLAHA, M., PREMERLANI, W., EDDY, F., & LORENSEN, W. (1991). *Object-oriented Modelling and Design.* Englewood Cliffs: Prentice-Hall.

SCHNEIDER, W., & SHIFFRIN, R.M. (1977). Controlled and Automatic Human Information Processing I, *Psychological Rev., 84,* 1-66.

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

SHEPHERD, A. (1989). Analysis and training in information technology tasks. In D. Diaper (Ed.), *Task Analysis for Human Computer Interaction* (pp. 15-55). Chichester, England: Ellis Horwood.

WALLACE, M.D., & ANDERSON, T.J. (1993). Approaches to Interface Design. *Interacting with Computers, 5* (3), 259-278.

WILLOCH MANAGEMENT CONSULTANTS (1993). *Vinnande Arbeidsprocesser og Organisationsstrukturer i Forandring* [Business Process Reengineering.]. Jar, Norway: Author.

ÅBORG, C., SANDBLAD, B., & LIF, M. (1996). *A practical method for Evaluation of Human-Computer Interfaces* (Rep. No. 75, CMD). Uppsala, Sweden: Uppsala University.

Appendix A: Definitions

The meaning of concepts in this area of research vary greatly depending on use. Therefore we chose to present the definitions we use and also introduce a few new concepts to fully understand the method and its advantages.

Goal - An intended objective (ISO/DIS 9241:11, 1995).

Task - Activities undertaken to achieve a goal (ISO/DIS 9241:11, 1995).

User - the person who interacts with the equipment (ISO/DIS 9241:11, 1995).

Actor - represents what interacts with the system. An actor represents a certain role that the user can play. (Jacobson, Cristerson, Jonsson, & Övergaard, 1992). One user can play the role of one or several different actors.

Judgment - "the mental or intellectual process of forming an opinion or evaluation of discerning and comparing". The capacity for judging is "the power or ability to decide on the basis of evidence" (Webster's 3rd New International Dictionary). Judgment is defined as a cognitive or intellectual process in which a person draws a conclusion or an inference about something that cannot be seen on the basis of data that can be seen (Arkes & Hammond, 1986).

Decision - "the act of settling or terminating...by giving judgment" (Webster's 3rd New International Dictionary). Decision is a choice between alternative action strategies. Here, we define a *decision* as made first when it is documented (Gulliksen, 1996a). A *judgment* is the process immediately preceding the momentary decision.

Variable - units of information needed to make a decision.

Features - Properties of the variables normally not found in the data model, such as status, frequency of use, type, and so on, of special importance to the user.

Actions - Operations a user undertakes to interacts with a system, such as, "information search", "enter a personal ID".

Work task - A continuos moment of work performed in order to reach a specific goal. Each work task includes a judgment process and is terminated by a decision.

Work situation - a set of related work tasks without sequential restriction, but with a natural belonging performed in total by one person. It corresponds to a workspace in the user interface. One work situation may include one or several work tasks. One actor can handle one or several work situations.

Work activity - describes all work that takes place within an organization or a domain of work.

Workspace - the interface that contains the information entities and the tools needed to fully perform one specific work situation