III

Workspaces enhance efficiency - Theories, concepts and a case study

MAGNUS LIF

EVA OLSSON

JAN GULLIKSEN

BENGT SANDBLAD

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

Abstract

Traditional process-oriented system development methods often results in fragmentary user interfaces with information presented in various windows without considering requirements for simultaneous viewing. Opening, closing, moving and resizing these windows attracts attention away from the actual performance of the work.

User interface design according to the workspace metaphor could provide skilled professional users with an efficient, customised user interface to administrative information systems. This can improve work performance and facilitate efficient navigation between workspaces. A case study in co-operation with the Swedish National Tax Board (RSV) describes practical use of the workspace metaphor.

1. Introduction

In the recommendations by the International Standardisation Organisation (ISO 9241-11, 1995), *usability* is defined as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. A general problem in the design of usable

interfaces concerns how a large and complex information structure can be visualised and controlled efficiently on a relatively small computer screen. A common solution to the limited screen space problem is to divide an application into a number of different windows, often hierarchically structured. The prevalent software development methods are probably promoting this application structure. A hierarchical application structure causes navigational difficulties, the user easily gets lost in information space (Woods & Watts, 1997). Interfaces where the users have to spend a lot of time rearranging windows instead of doing their actual work are unfortunately common (Gulliksen & Sandblad, 1995). To be able to design an efficient interface it is essential to understand that the users' main interest is to perform their work tasks, not to operate a computer. The computer system must support the user with appropriate sets of information that can be reached fast and with minimal cognitive effort.

Skilled users are most often forced to use one particular support systems and the efficiency of their work is strongly related to the usability of that system. Badly designed support systems can contribute to cognitive as well as psycho-social work environment problems (Smith & Carayon, 1993). Our research aims at identifying cognitive work environment problems, and to find solutions to them. Work environment problems are often specific to certain work domains. The design must therefore be based on extensive domain knowledge in order to support each category of users with a tailored user interface (Gulliksen & Sandblad, 1995). One of the domains we have studied more carefully is the domain of case handling (Gulliksen, 1996) in administrative environments, such as banking, insurance or public service work. In the administrative domain users often require simultaneous access to a lot of information to be able to make a proper decision. The *design* of the user interface is especially important when developing computer support for skilled professionals (Nygren, Johnson, Lind, & Sandblad, 1992).

Two main approaches to design will be discussed, a process-oriented and a workspace-oriented approach. Our hypothesis is that user interfaces based on workspaces efficiently can support skilled user, especially in administrative work domains.

2. Process-oriented versus Workspace-oriented Design

2.1. A process-oriented approach

Traditional software development is mainly performed with a process-oriented approach. The users' work is typically specified with a data model and a definition

of the processes, e.g. through a data flow diagram (DeMarco, 1978). Each process often corresponds to one function of the planned application. In such methods there is seldom any suitable aid provided for developing dialogue interfaces (Floyd, 1986), i.e. how to use these models in the interface design process. Recent object-oriented methodologies, such as UML (Booch, Jacobson, & Rumbaugh, 1997), do not give enough support for this process either. Instead, the process-oriented approach invites the designer to create an interface where each specified function corresponds to one or more windows. Typical modelling and design work consist of the following steps:

- Specify the functions that involve interaction with a user.
- Define a structure for how the different functions can be accessed from each other during the work process.
- Give each function its own, unique user interface (window).
- Create a menu on the top level from which the user can access the functions needed for completing the work process.

In a process-oriented structure the application is the sum of all functions of the application (Figure 1). The user interface contains *all* functions that *all* different users can access in *all* different work situations. A user works with the application by selecting among the desired functions in a menu or by navigating through the structure of functions. Each function will have the same look and behaviour independent of who accesses it, for what purpose it is accessed, and in which context it is accessed.

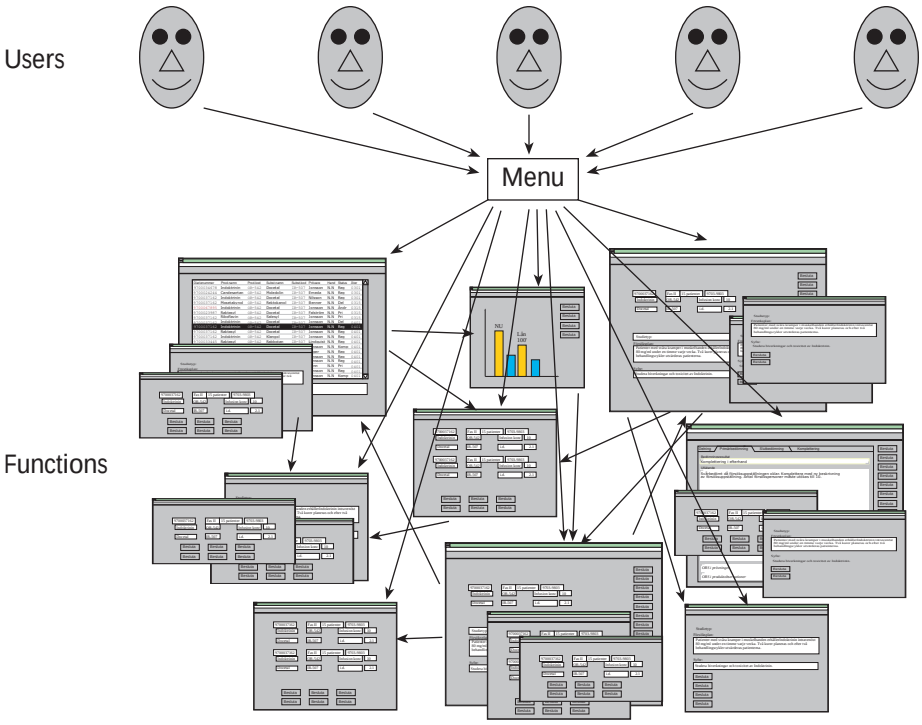

Figure 1. The process-oriented approach often results in a fragmentary interface containing several windows corresponding to the different functions.

There are some major problems associated with this approach:

- the design is not adjusted to the users actual tasks
- the user has to spend time on opening, closing, moving and rearranging windows while performing the task
- the user often has to consult several different functions in order to complete a certain task

In order to reach the desired functions a user must understand and navigate through a rather complex structure of windows and menus. The described way of working is inappropriate because parallel presentation of information is more efficient than sequential presentation (Lind, 1991).

It is easy to proceed from the process oriented analysis to a multiple window design and the result will most likely be a fragmentary interface split into several windows. Furthermore, to deal with an assignment a user typically has to use more than one

application, where each application has its own main window with accompanying secondary windows. In a large corporation with in-house development, each application might in addition have its own unique look and feel. In order to overcome these problems a more appropriate metaphor has to be chosen.

2.2. A Workspace-oriented approach

With a workspace-oriented approach the interface is designed to support categories of users in their different work situations (Lif, 1997). The main idea here is to support each category of users with a *tailor-made* interface, that is complete with respect to information contents and tools for each of all the different work situations in which the users can be involved.

Modelling and design according to the workspace-oriented approach means specifying, among other things, the following:

- The different *categories of users* in the organisation.
- The *work situations* these categories of users will be involved in.
- The *information and the tools* that are needed in each work situation.
- The *interaction requirements* in each work situation, as a basis for design of usable interfaces.

The workspace-oriented approach results in a design that is specifically tailored according to how the professionals actually perform their work procedures. Each work situation corresponds to a workspace on the screen. A workspace is the user interface to a complete work situation. On the top level of the system, the relevant workspaces are made available for each category of users via a task panel. When a user has selected a workspace, the screen is ready to use, without the need for further design actions, with respect to information and tool contents (Figure 2). The number of different workspaces (work situations) for a specific user is often relatively small, less than five or six should be sufficient for a user in the domains we have studied.

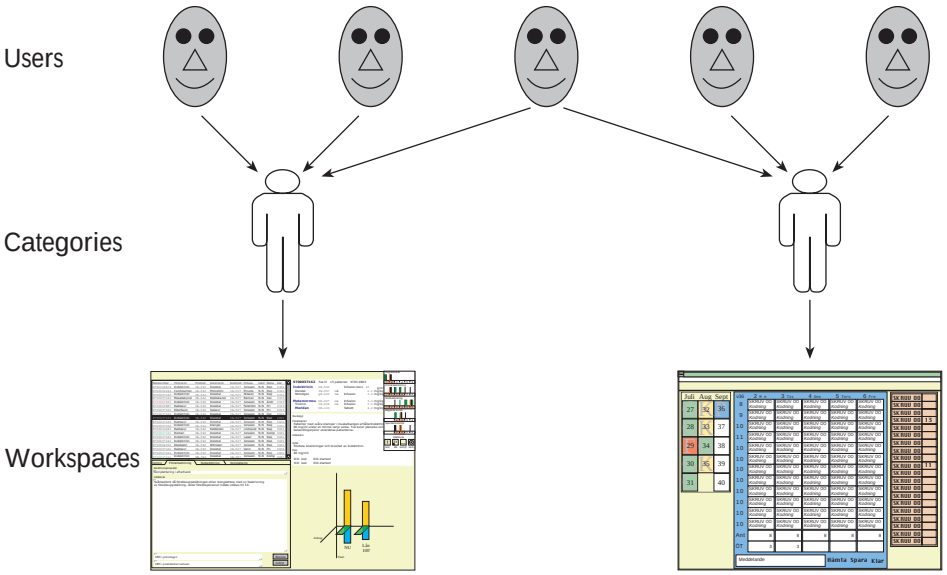

Figure 2. Each category of users can potentially perform work in one or more different work situations. Each work situation corresponds to a workspace, i.e. an interface to exactly the part of the information system that supports the work situation.

Administrative work, such as case handling, often requires an overview of an extensive amount of information. The interface design must be as complete as possible in order to satisfy demands on simultaneous presentation of information. When all required information and all operations are gathered in one workspace it is easier for the user, frequent as well as novice, to immediately become oriented in the information space and get support for further actions. In addition each workspace must have a distinct layout well separated from others. This will help the user to immediately recognise the workspace, its contents and the included operations, when switching to another task. The workspace concept, as described here, would not only support the user while performing each work task, but also provide an overview and facilitate task switching.

We mean that using a workspace oriented approach to design offers better possibilities to create an interface that supports the users in their work. We will now, in more detail, describe the workspace metaphor as an alternative to "multiple windows".

3. The Workspace metaphor

The workspace metaphor is based on a framework for a multiple virtual workspace called the *rooms design* (Henderson & Card, 1986; Card & Hendersson, 1987). The idea behind the rooms design is to provide the user with a number of screen-sized workspaces called rooms. Each room has a set of small icons ("doors"), used for navigation between rooms. Each room contains a number of windows that support the task to be performed in that room. A modified version of the rooms design is here suggested to replace the prevalent desktop metaphor for administrative information systems.

3.1. Work task and actor

An important concept related to the workspace metaphor is the work task. A work task in this context is defined as a continuous moment of work performed in order to reach a specific goal. Each work task includes a judgement process and is terminated by a decision. The interface must be structured according to the actual work tasks and how the users perform them. The analysis of the work tasks that the application will support must not only cover all different aspects of the work to be done, but also identify the different actors and which tasks they perform. For a more thorough analysis of the concepts concerning work tasks and actors see Gulliksen, Lif, Lind, Nygren & Sandblad (1997).

3.2. Workspace

Each workspace has to completely support one or more work tasks in order to relieve the user from having to switch to another workspace during a judgement process. A workspace can be thought of as a container enclosing the interface elements that hold the information needed for a work task. One specific interface element can appear in different workspaces since the same information, possibly viewed in different ways, may be required in many tasks. An element can for instance be presented in one workspace where the information is read-only and in another workspace the same information may be editable. Which tasks that can be performed in one workspace depends on their logical relations and their connections in time. An administrative work task can for instance consist of the following actions:

- taking care of a newly arrived document
- finding out that this document concerns a request for delay of the current case

- analysing the arguments for the delay, checking current status and previous history
- make a decision about the request, accept it, reject it or request more information from the concerned part
- document the decision

This work task has then been completed and the next task can be handled. Depending on the nature of other assignments connected to the same kind of case, additional tasks could be performed in the same workspace as well. The work is seldom performed in a perfect sequence from start to finish. Therefore, the user must always be able to interrupt and leave an ongoing work task in one workspace, and return later to complete the task.

3.3. Actor - workspace

The concept of actor (Jacobson, Christerson, Jonsson, & Övergaard, 1992) is used to describe different categories of users. Each workspace is specially designed to fulfil the requirements of a certain actor and has consequently a specific authority level. The contents and the layout will depend on the requirements of the intended actor. This implies that a workspace may admit or deny access to (show or hide) information depending on the authority of the current actor. A specific user can play the role of one or more actors. An actor will typically have access to a number of workspaces and use a few of them frequently.

The relations between the defined concepts are many-to-many, i.e. one actor can be engaged in several work situations, one work situation can consist of several tasks. One task can as well be a part of different work situations.

4. Designing a workspace

The basic idea with the workspace metaphor is that a professional user should be able to perform an entire work task in one workspace. The completion of a single task should not require switching between different workspaces. Each workspace must be carefully designed to support the users in their work. We will here introduce some design heuristics to support the design of the workspace interface. The heuristics are especially important when designing according to the workspace metaphor, but most of them are also valid when designing for skilled users according to other metaphors.

4.1. Simultaneous presentation of information

During decision making, humans tend to omit information that is not immediately available (Tversky & Kahneman, 1974). A skilled user can easily overview a large set of familiar data (Nygren & Hendriksson, 1992). To be able to make a proper judgement, the information must be simultaneously visible on the screen. In our experience it is possible to present more information on the screen by using information coding and a careful spatial layout. Window-based information systems almost always occupy the screen with information irrelevant for the skilled professional, such as window frames. Windows tend to cover important information in other windows. Information presented on screens that can only be viewed in sequence has proved to be more cognitively demanding than information presented in parallel (Lind, 1991). The process of keeping decision relevant information in the short-term memory is often disturbed when manoeuvring through the information space. Due to the advantages of simultaneously presented information, hierarchies should be avoided.

4.2. Support pattern recognition

Experienced users decode frequently occurring, meaningful patterns quickly (Nygren, Allard, & Lind, In press). If a set of variables always has the same properties (such as colour or shape and spatial location on the screen) global patterns may emerge over time that can be used to guide the users in the reading process. The decoding of patterns can be performed on an automatic cognitive level without interfering with processes performed on a conscious level.

4.3. Workspace contents

Each workspace must be especially equipped to supply the user with all information and every tool needed to perform all parts of the work task. This will render each workspace an obvious character that immediately reveals the kind of tasks that can be performed there, which facilitates recognition of the different workspaces. Some generic tools have to be available from any workspace (such as electronic mail, calculators, note pads), depending on the kind of activities performed in the current environment.

5. A Case study

In our co-operation with the Swedish National Tax Board (RSV) the workspace metaphor has been implemented to facilitate case handling. Characteristic for case handling is that a lot of information from different sources has to be gathered and verified to enable a correct decision. Traditionally, the users have performed their

work by interacting with several different applications. The purpose here was to create one application where each user has access to different workspaces tailored for each work situation, as defined by the workspace metaphor. The application was designed for a PC with a 17-inch screen (resolution = 1024*768). Different workspaces were designed and implemented by different groups of software engineers. The choice of development tool differed between the groups.

In the analysis of the users' work, a number of work situations that has to be supported by the future system were identified. Examples of such work situations are: detecting income tax returns that must be reviewed; examine income tax returns and deciding the customers' final tax. Each work situation should correspond to a workspace in the user interface. The users have different authorities, therefore the individual number of required workspaces differ.

Figure 3 shows a typical workspace included in the system. All information and every tool needed for the judgement and decision-making process are included in the workspace. Every interface element included in the workspace has its fixed position on the screen. On the left side of the display a ledger is shown containing information about all previous events and decisions concerning a case. This ledger provides a direct index to previous decisions. An overview is shown for the case as well as facilities for making a decision on how to handle the current matter.

Switching between workspaces is enabled through a task panel (to the upper right), either permanently visible, or hidden and occurring upon demand. The contents in the task panel depend on the users' work situation. Each workspace can be selected directly via the task panel. A workspace is represented with an icon and a short explanation (tool tip).

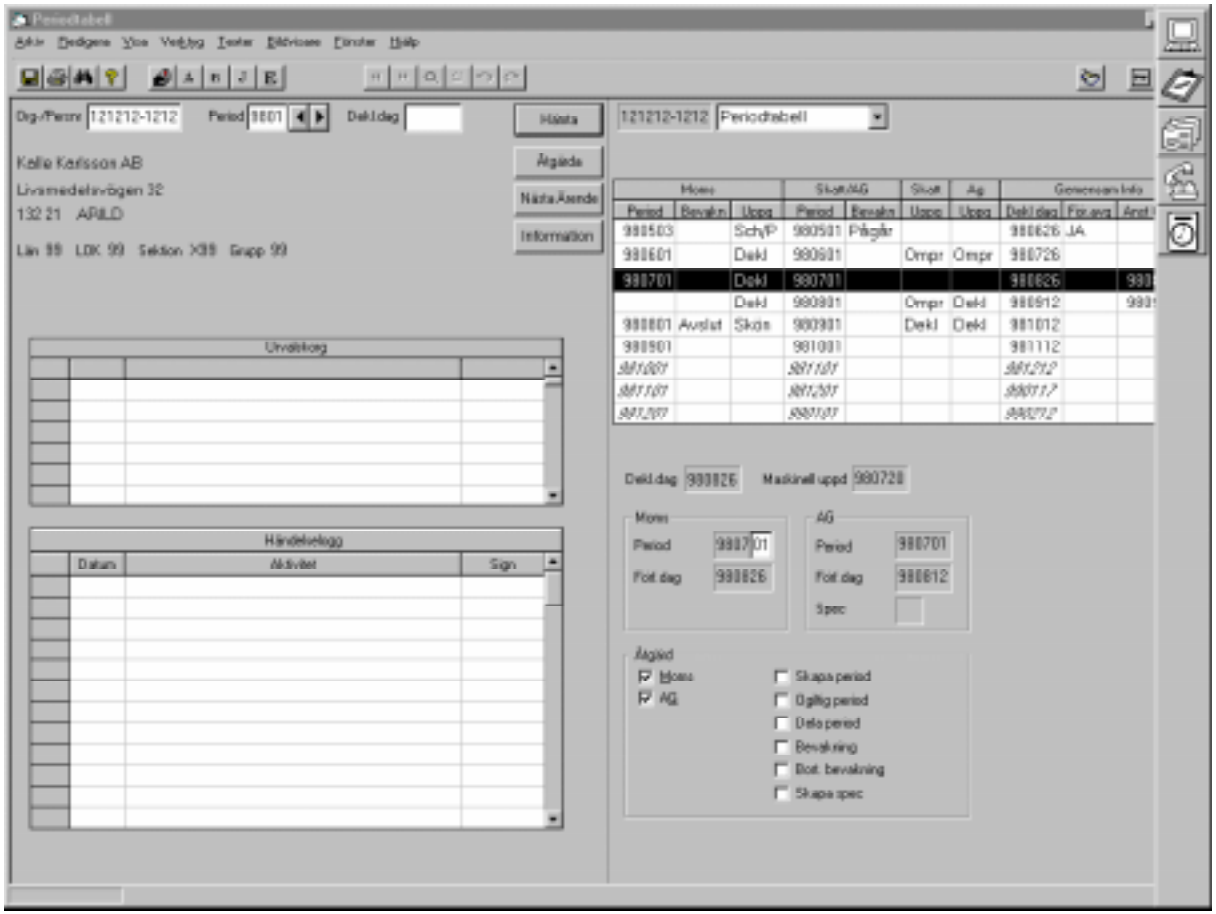

Figure 3. The display shows a selected workspace. The task panel that contains an overview of available workspaces is visible to the right. (Colour plate 3).

5.1. Method

The purpose with this case was to study the implementation of the workspace metaphor at the Swedish National Tax Board. The workspace metaphor has been implemented in five different projects and in two of these we have been engaged as usability experts, giving advice on how to design parts of the user interface. The first version of the system was evaluated with a *heuristic* approach (Nielsen & Molich, 1990). Three HCI experts, with limited domain knowledge, and five work activity experts, evaluated the user interface. Each evaluation was performed individually. The heuristics used in the evaluation were selected by the person responsible for usability issues at the Swedish National Tax Board in co-operation with the authors.

5.2. Results

Some problems related to the implementation of the workspace metaphor were identified in the evaluation. The most important findings are listed below:

- *Task switching.* Sometimes it is necessary to interrupt the on-going work before a decision has been made and initiate a new task, e.g. when a customer calls and needs immediate service. In the current system a new case usually can not be initiated unless the current case is either completed or postponed.

- *Mainframe systems.* Getting on-line access to and change information stored in older mainframe systems is a problem. The user has to start a new application placed in a separate window on top of the current workspace. The contents of this window is not graphical and is handled differently than the ordinary workspace. If the user clicks outside this window, it will be placed behind the current workspace, which might be confusing for infrequent users, and problematic to recover from.

- *Lack of information.* There were several complaints about the need for more information on the screen. Sometimes important information is hidden and sometimes it is not even possible to view the required information without cancelling the current work task.

- *Consistency.* The way of switching between workspaces is not consistent with the way it is done in the Win95™ task bar. That was considered as a potential usability problem.

- *Workspace character.* Most of the evaluators found that it was difficult to recognise the kind of work tasks that could be performed in a workspace.

Furthermore, the layout of many of the workspaces was too similar, making them difficult to distinguish.

5.3. Comments on the results

Some of the problems described above can be explained by the difficulties in mixing old systems with new ones. Placing information from the older system in a separate window, on top of the current workspace, will most likely lead to problems since the user then is unable to view all needed information simultaneously when making a decision. Furthermore, opening a new secondary window each time it is placed behind the main window can eventually cause system overload.

Other problems concern the analysis of the users' work. The analysis has to be very thorough in order to avoid usability problems later on (indeed this should be the case independent of the choice of system metaphor). This puts immense demands on the analysis team in order to reach a level where all needed information, for certain is supplied within a workspace. An incomplete analysis could for instance cause a situation where the user has to switch between workspaces in order to perform a task, which makes the produced workspaces just as inadequate as an application using multiple windows.

Five separate but interrelated development projects were started before the actual workspaces were identified. The different projects teams developed applications in parallel. When the workspaces were specified we discovered that the separation into smaller applications was inappropriate since information in some workspaces had to be fetched from local databases included in other applications.

The result of the evaluation performed at RSV shows that the screen space seldom is perfectly organised. Much of the needed information is hidden in dialogues, tab controls or displayed on the users' request. Implementing the workspace metaphor puts new demands on the software engineers. It is difficult for developers familiar with the ordinary desktop with multiple windows to immediately adapt to the workspace metaphor. Nielsen et al. (1992) have discussed similar problems in a survey where software engineers were taught object-oriented design. They conclude that experience is needed and advises from specialists are useful. We believe that the workspace metaphor could be adopted more readily if the developers were supported by methods for analysis that capture aspects directly applicable in the design of the user interface.

The problems detected in the evaluation points out potential pitfalls in the implementation of the workspace metaphor in a computer system. Most of them occurred because of technical limitations, insufficient task analysis and difficulties in understanding the users' needs. They did not depend on the workspace metaphor itself. So

far we have only been able to study the use of the workspace metaphor in a first version of the system. Some of the detected problems will be dealt with in later versions. In the future we will see the effects of long time use.

6. Discussion

The limited screen space is one of the major obstacles for being able to create a usable office work situation. The workspace metaphor represents an interesting approach to solve some of the basic design problems, such as task switching and visualisation of large and complex information structures onto the limited screen space. Based on basic principles of the rooms metaphor as presented in (Card & Henderson, 1987; Henderson & Card, 1986) we have developed a structure for a workspace oriented design methodology. Our ambition has been to create a framework for design that easily can be implemented using most commercial development tools.

A workspace design method must be a part of a complete methodological chain for systems development. In order to fully utilise the advantages of the discussed method, the design work should be based on an object oriented system development and data modelling method. The workspace metaphor is well suited to work in conjunction with the use-case approach (Jacobson et al., 1992) and User Interface Modelling (Lif, 1997).

The design of a workspace must be elaborate. It is not sufficient to make sure everything needed is there and leave the final layout to the user. This kind of design almost always ends up with a work situation where the user constantly has to rearrange the contents of the workspace in order to display the information needed for the moment. Frequently needed information must be structured in a way that facilitates retrieval. The design requirements stated here might seem contradictory compared to a changing work situation. Even though a perfected design is desired, it must at the same time be possible to let the application evolve in time and grow with the work task that inevitably will change over the coming years. A modular way of thinking is required, otherwise the application will soon be outdated. Consequently the objects in a workspace must be easy to replace and it must be possible to append new elements. New methods or services must be possible to add without disturbing the previous structure of a workspace.

An information system designed according to the workspace metaphor can work more efficiently if all of the users' applications are workspace oriented. Achieving this is a matter of supervisory status, a strategic issue. One way of accomplishing this is by writing the workspace metaphor into a corporate style guide which has been done at the Swedish National Tax Board. This can be done according to the theories

of domain specific design, that is with the use of a domain specific style guide as a basis for making the design decisions (Gulliksen & Sandblad, 1995; Borälv, Göransson, Olsson, & Sandblad, 1994). The objectives behind the development of a domain specific style guide are, among other things, to include more domain knowledge in the design of the user interfaces to applications in that domain.

The workspace metaphor has been applied in several research projects (e.g., Borälv et al., 1994; Borälv & Göransson, 1997). It has been used to a wider extent at the Swedish National Tax Board where descriptions on its application and functionality have been specified in the organisation's domain specific style guide.

The full implementation of the workspace metaphor in a large organisation, with requirements on high performance and compatibility with other systems, can be problematic. However, it has the potential of solving many of the basic problems encountered when designing complex computer support systems.

Acknowledgements

This work was performed with financial support from the Swedish Council for Work Life Research (RALF), the Swedish National Board for Industrial and Technical Development (NUTEK), and in co-operation with the Swedish National Tax Board (RSV). All co-operation and contributions from project members are greatly appreciated.

References

BOOCH, G., JACOBSON, I., & RUMBAUGH, J. (1997). *Version 1.0 of the Unified Modelling Language*, (On-line). Available: http://www.rational.com/uml/references/docset.html.

BORÄLV, E., & GÖRANSSON, B. (1997). A Teleradiology system. In *Proceedings of Designing Interactive Systems* (pp. 27-30). Amsterdam: ACM.

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, 45*, 63-76.

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.

DEMARCO, T. (1978). *Structured Analysis and System Specification.* New York: Yourdon Press.

FLOYD, C. (1986). A comparative evaluation of system development methods. In T.W. Olle, H.G. Sol, & A.A. Verrijn-Stuart (Eds.), *Information systems design methodologies: Improving the Practice* (pp.19-55). Elsevier Science Publishers B.V.

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* (pp. 225-256). Norwell, MA: Kluwer Academic Publishers.

GULLIKSEN, J., LIF, M., LIND, M., NYGREN, E., & SANDBLAD, B. (1997). Analysis of Information Utilisation. *International Journal of Human-Computer Interaction, 9* (3), 255-282.

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

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.

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.

LIF, M. (1997). User interface modeling for design of administrative information systems. In *Proceedings of the 7th International Conference on Human-Computer Interaction, HCI International' 97* (pp. 383-386). San Francisco, U.S.A: Elsevier.

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

NIELSEN, J., BUSH, R.M., DAYTON, T., MOND, N.E., MULLER, M.J., ROOT, R.W. (1992). Teaching Experienced Developers to Design Graphical User Interfaces.

In *Proceedings of ACM CHI' 92 Conference on Human Factors in Computing Systems, The Role of the Organization in System Design* (pp. 557-564). Reading, MA: Addison-Wesley Publishing Company, Inc.

NIELSEN, J., & MOLICH, R. (1990). Heuristic evaluation of user interfaces In J.C. Chew, & J. Whiteside (Eds.), *Proceedings of Human Factors in Computing Systems, CHI' 90* (pp. 249-256). New York, NJ: ACM.

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.

SMITH, M., & CARAYON, P. (1993). A balance model for examining psychological stress in VDU work. In H. Luczak, A. Cakir, & G. Cakir (Eds.), *Work with display units -92* (pp. 35-39). Amsterdam: Elsevier Science Publisher.

TVERSKY, A., & KAHNEMAN, D. (1974). Judgement under uncertainty: Heuristics and biases. *Science, 185*, 1124-1131.

WOODS, D.D., & WATTS, J.C. (1997). How Not to Have to Navigate Through Too Many Displays. In M. Helander, T.K. Landauer, & P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 617-650). Amsterdam: Elsevier Science B.V.