Human Work Interaction Design: Designing for Human Work

Edited by
Torkil Clemmensen
Pedro Campos
Rikke Orngreen
Annelise Mark Pejtersen
William Wong

Springer

ifip

HUMAN WORK INTERACTION DESIGN: DESIGNING FOR HUMAN WORK

IFIP – The International Federation for Information Processing

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP's aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

> *IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.*

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP's events range from an international congress to local seminars, but the most important are:

- The IFIP World Computer Congress, held every second year;
- Open conferences;
- Working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is less rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.

HUMAN WORK INTERACTION DESIGN: DESIGNING FOR HUMAN WORK

The first IFIP TC 13.6 WG Conference: Designing for Human Work, February 13-15, 2006, Madeira, Portugal

Edited by

Torkil Clemmensen
Copenhagen Business School

Pedro Campos
University of Madeira

Rikke Orngreen
Copenhagen Business School

Annelise Mark Pejtersen
Center of Cognitive Systems Engineering

William Wong
University of Middlesex

Springer

Library of Congress Control Number: 200692674

Human Work Interaction Design: Designing for Human Work

Edited by T. Clemmensen, P. Campos, R. Orngreen, A. Mark Pejtersen, and W. Wong

p. cm. (IFIP International Federation for Information Processing, a Springer Series in Computer Science)

ISSN: 1571-5736 / 1861-2288 (Internet)
ISBN: 10: 0-387-36790-X
ISBN: 13: 9780-387-36790-X
eISBN: 10: 0-387-36792-6
Printed on acid-free paper

Printed in the United States of America.

9 8 7 6 5 4 3 2 1
springer.com

Contents

The International Federation for Information Processing –IFIP

IFIP is a non-governmental, non-profit umbrella organization for national societies working in the field of Information and Communication Technologies. It was established in 1960 under the auspices of UNESCO as an aftermath of the first World Computer Congress held in Paris the previous year. IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of Information Technology for the benefit of all people.

IFIP's manifold activities include academic publications and research events that range from workshops and working conferences to global conferences such as the bi-annual World Information Technology Forum (Witfor) and the well established World Computer Congress (WCC). Events and publications are mainly organized by a series of Technical Committees, which manage more than 3500 scientists from Academia and Industry. The scientists operate in one or more of the Technical Committees as well as in the Working Groups and in the Special Interest Groups that are established within each particular Technical Committee to which they also report their activities.

See http://www.ifip.org/ for aims and scopes of each committee and associated Working Groups and Special Interest Groups.

Publications and proceedings from IFIP events are published by Springer Verlag in its IFIP series.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country.

1.1 Technical Committee TC13 on Human – Computer Interaction

These committees also include the Technical Committee TC13 on Human – Computer Interaction within which the work of this volume has been conducted. TC 13 on Human-Computer Interaction has as its aim to encourage theoretical and empirical human science research to promote the design and evaluation of human-oriented ICT. Within TC 13 there are different Working Groups concerned with different aspects of Human-Computer Interaction.

The flagship event of TC13 is the bi-annual international conference called INTERACT at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

Publications arising from these TC13 events are published as conference proceedings such as the INTERACT proceedings or as collections of selected and edited papers from working conferences and workshops.

See http://www.ifip.org/ for aims and scopes of TC13 and its associated Working Groups.

1.2 Working Group 13.6 on Human-Work Interaction Design

This Working Group was established in September 2005 as the sixth Working Group under the TC13 on Human - Computer Interaction. It focuses on Human-Work Interaction Design (HWID) and it is called WG13.6. A main objective of the Working Group is the analysis of and design for a variety of complex work and life contexts found in different business and application domains. For this purpose it is important to establish relationships between extensive empirical work-domain studies and HCI design. The scope of the Working Group is to provide the basis for an improved cross-disciplinary co-operation and mutual inspiration among researchers from the many disciplines that by nature are involved in a deep analysis of a work domain. Complexity is hence a key notion in the activities of this working group, but it is not a priori defined or limited to any particular domains. It is our hope that this Working Group on Human-Work Interaction Design (HWID) will also lead to a number of new research initiatives and developments, as well as to an increased awareness of HWID in existing and future HCI educations.

http://www.ifip.org/bulletin/bulltcs/memtc13.htm#wg136

Foreword

This book records the very first Working Conference of the newly established IFIP Working Group on Human-Work Interaction Design, which was hosted by the University of Madeira in 2006. The theme of the conference was on synthesizing work analysis and design sketching, with a particular focus on how to read design sketches within different approaches to analysis and design of human-work interaction. Authors were encouraged to submit papers about design sketches - for interfaces, for organizations of work etc. - that they themselves had worked on. During the conference, they presented the lessons they had learnt from the design and evaluation process, citing reasons for why the designs worked or why they did not work. Researchers, designers and analysts in this way confronted concrete design problems in complex work domains and used this unique opportunity to share their own design problems and solutions with the community.

To successfully practice and do research within Human - Work Interaction Design requires a high level of personal skill, which the conference aimed at by confronting designers and work analysts and those whose research is both analysis and design. They were asked to collaborate in small groups about analysis and solutions to a common design problem.

The response to this very first HWID conference was positive in terms of submissions and participation. We received fifteen long papers and six short papers and selected 70% of the submitted papers to be presented at the conference and published in the proceedings. The interest shown in the conference was truly international; we had submissions from 10 countries in Europe, Asia and North America. The result is a set of innovative papers that challenge the relationship between design sketches and work analysis from different perspectives.

Organising this first 13.6 Working Conference from the first call for papers through the actual conference activities till the final publication of the papers is not a simple task. It required hard work and commitment from both those who were involved in the intellectual calls and reviews and the practical work of setting up the environment for the conference as well as from the participants during the conference. We want to thank everybody for their significant contribution to the success of this conference. A special thanks to Nuno Jardim Nunes, the University of Madeira. We also want to thank IFIP and the sponsors of the conference for providing the support and funding for its implementation. The sponsors are IFIP, FLAD (the Luso-American Foundation) and FCT (the Portuguese Foundation for Science and Technology).

April 2006

Annelise Mark Pejtersen
Chair, IFIP WG 13

Torkil Clemmensen
Vice chair, IFIP WG 13.6

Activity Theory for Design

From Checklist to Interview

Matthew Duignan, James Noble[1], and Robert Biddle[2]
1 School of Mathematics and Computing Sciences
Victoria University of Wellington, New Zealand
Matthew.Duignan@mcs.vuw.ac.nz

2 Human Oriented Technology Laboratory
Carleton University, Ottawa, Canada
Robert_Biddle@carleton.ca

Abstract. Cultural historical activity theory has shown much promise as a framework for Human Computer Interaction, particularly for analysing complex activity and its context. However, it provides little practical methodological support for user interface designers. This paper presents an activity interview resource which can be used by interface designers when developing new tools to support creative activity. The new activity interview is based on the excellent foundation of the activity checklist, but resolves a number of its deficiencies. In particular it provides concrete questions to fuel an activity theory analysis, rather than the more abstract and less accessible checklist. We describe how we have dealt with these problems, and reflect on our experience applying the activity interview in the domain of computer mediated music production.

2 Introduction

Supporting creative computer-based activity presents many challenges to interface designers. Its open-ended nature requires non-prescriptive and flexible designs that still meet the needs of creative professionals. Understanding these needs requires a thorough and holistic analysis of their activities. We have been conducting research of one such activity: computer meditated music production. Our aim is to challenge existing user interface metaphors, and design new and compelling alternatives. In our work we found a need for a framework and methodology for analysing this activity, and turned to cultural historical activity theory [16, 10, 19].

Activity theory has been gathering recognition as a useful analytical framework for understanding human activity. However, it provides little of the crucial methodological support required for practical use in design. One resource that has worked towards meeting this need is the activity checklist [13]. We found that while an invaluable resource, the activity checklist has some significant shortcomings.

In this paper we present a practical research tool we have developed for use in the design process. It provides a series of interview questions designed to lead

Please use the following format when citing this chapter:

Duignan, M., Noble, J., Biddle, R., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 1–25.

directly to an activity theory analysis. This tool is based on the activity checklist, but attempts to address some of its limitations. In particular it provides concrete questions to fuel an activity theory analysis, rather than the more abstract and less accessible checklist.

Firstly we show the motivation for such a tool. Next we provide a brief primer on activity theory and its use in Human Computer Interaction (HCI). We introduce the activity checklist and present our critique of its limitations and then present our activity interview resource. Finally we reflect on our use of this new resource in the domain of computer mediated music production.

3 Motivation

Computer mediated music production is just one of a class of open-ended creative activities supported by computer software. Many of these activities can be found in the various forms of multimedia development. Two and three dimensional modelling and animation, graphic design, interactive art, computer game modelling, and some forms of industrial design have many important features in common.

These activities are typified by several characteristics. Subjects' actions in these activities are non-prescriptive and they do not follow a predefined or predictable ordering. Different actions are carried out as and when needed and in response to the emerging requirements of work. Moreover, these activities are highly open-ended. While subjects may have some general idea of what they wish to achieve, the specifics of the result will depend entirely on the unfolding and unpredictable nature of their creative process.

We have been working on a detailed analysis of the activity of computer mediated music production. Here we use the term production to include the entire process of composing, recording, editing, arranging, and mixing of musical content. We also include the process of preparing material for use in a live context. Our analysis is informed both by the activity interview which we present here, cognitive dimensions analysis through interviews [6], and observations. The resulting analysis has provided us with many useful insights into this activity and is informing our design of a novel user interface which breaks some of the music production metaphors that currently dominate music interface design. We will continue by producing prototypes and design sketches. Figure 1 shows the Arranger and Session views of the Ableton Live [1] music production software (above) and one of our very early prototypes (below). Ableton Live (and most other audio and midi sequencers) embodies strong multi-track and mixer metaphors. We are working towards a sequencer which allows for custom abstractions which retain the conceptual clarity of multi-track and mixer models, but allows more flexibility of structural expression.

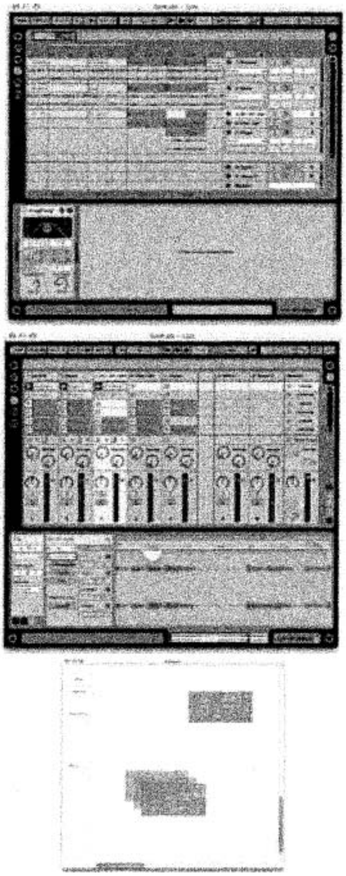


Fig. 1. Ableton Live's Arranger and Session views (above) and one of our early prototypes (below).

In music making and other similar activities contextual concerns can have large consequences for how software is used. There is typically a large degree of collaboration between the various subjects who take part in music production. These subjects include producers, composers, instrumentalists, and engineers. They often need to share the results of their work, both in real-time, and after the fact. Due to the aforementioned open-ended nature of this activity, it is impossible to prescriptively describe the ways in which any given tool will be used in collaboration.

Different subjects will have different skill specialisations, and hence use different tools which may provide obstacles to their integration. Even single subjects often use a complex and heterogeneous tool set. Any difficulties in getting these tools to work well together in a manner that complements the activity can be a major impediment to creativity and efficiency.

The study of HCI has lead to the development of methodologies for analysing many types of task. However techniques such as task analysis and decomposition fall short when applied to these more rich and dynamic domains. Traditional methods typically fail to account for "the tacit knowledge that is required in many skilled activities, or the fluent action in the actual work process" [2] which need to be taken into consideration when analysing open-ended and exploratory activities. These methods can also fail to describe the rich context of activities such as music production, which must be taken into account when looking at tool design.

One analytical framework that has been claimed to have strengths in describing context and creative action is cultural historical activity theory [16].

4 Background on Activity Theory

Activity theory stems from the Russian branch of Psychology, based principally on the work of Vygotsky which emerged in the 1920s [19]. The resulting family of theory focuses on the context of human activity, and explains human consciousness in these terms. The theory uses the notion of an "Activity" as its central unit of analysis. Activity is a high level of analysis which includes the meaningful context of individual actions in a group context, rather then just the actions themselves. It can be difficult to clearly define activity theory due to the large quantity of competing and complementary theory that has developed under its banner over the years.

Nardi argues that activity theory has much to offer the process of user-interface design as it:

> "... offers substantial tools for a broadly scoped study of `computer-mediated activity'... it weaves together, in a single coherent framework, so many interesting theoretical constructs crucial to an understanding of human activity: dynamic levels of activity, mediation, contradiction, intentionality, development, history, collaboration, functional organ, the unity of internal and external." [16](p.375)

Activity theory encompasses and integrates a number of different areas of analysis for any given activity, including structure, context and development.

The structure of the activity is decomposed into its sub-components, and analysed in terms of human motivations. This decomposition includes the full range from understanding the broad activity and its motives, individual conscious actions and goals, down to people's sub-conscious operations and the conditions that trigger them [14].

In activity theory, subjects' consciousness is explained through their internalisation of abilities such as mental addition, mental simulation, touch typing, and language. Humans also extend and share their consciousness through externalisation. This includes note taking and the use of symbolic and abstract tools.

The context of an activity is analysed in terms of various influencing forces in the environment in which the activity takes place. Activity theory places a special emphasis here on how tools mediate action between any number of subjects and an object. The contextual analysis also examines the impact of the surrounding community, explicit and implicit rules, and the division of labour.

These various aspects of the theory provide a broad and flexible framework and terminology for describing human activities and the factors that influence them.

5 Activity Theory in HCI

Much of the comparatively recent research in activity theory has focused on activity systems in complex work environments [10]. This led firstly to a natural growth in its application to computer-supported collaborative work (CSCW), and later to further use in the more general field of HCI. In 1991 Bødker introduced activity theory to HCI explicitly in *Through the Interface* [7].

Activity theory seems to be a natural fit in the HCI domain due to its basis in the central concept of the mediating role of tools in human work. Activity theory provides a comprehensive framework for many of the concerns of HCI practitioners and researchers:

- The understanding of activity structure as a hierarchic set of "activity, actions, and operations" provides additional conceptual insight to conventional HCI techniques such as hierarchical task decomposition.
- Activity theory's model of the social environment provides a framework for the HCI concern of ensuring an interface takes into account contextual issues.
- The study of historical development in activities can provide HCI experts with a model for understanding how computer tools have changed activities in the past, and how changes they make may impact the future activity.
- Activity theory highlights the inherent contradictions in activities, which must be taken into account in interface designs which support these activities.
- User interfaces need to facilitate the natural human processes of internalisation of activity, and externalisation for extension of consciousness. Activity theory provides a theoretical framework to account for these processes.
- Finally, Activity theory exposes the importance of the transition between conscious and subconscious cognition, which is of great importance for learning and mastery of user interfaces.

Most importantly, activity theory provides a body of thought and unified framework for these concerns that might have previously fallen under the term "task analysis". Activity theory can provide this framework in concert with traditional methods such as user centred design [17]. This allows the HCI specialist to develop a thorough and broad understanding of an activity, and a consistent model to describe it.

However, activity theory's long history, wide-ranging complex and rich approach, and theoretical focus has not lent itself to easy application in the HCI field. Activity theory provides no step-by-step methodology.

Some HCI practitioners have taken a general activity theoretic approach to HCI analysis by firstly immersing themselves in the varied activity theory literature, and then using other techniques to learn about a specific domain where they wish to develop a new user-interface. Activity theory is then used as a framework for explaining and making sense of their findings [16]. This sort of model can then be used as the basis for further, more systematic analysis. For example, Turner and Turner [18] carefully uncovered activity theory "contradictions" in a domain to drive requirements gathering.

While these general activity theoretic approaches have their place, mastering the entire framework is an onerous task which has likely held back its application more universally. With no clear methodological guide it is difficult for HCI practitioners to take advantage of the framework's potential.

Nardi et al. [13], some of the chief theorists behind activity theory's application to HCI, admitted that:

> "These general principles [of activity theory] help orient thought and research, but they are somewhat abstract when it comes to the actual business of working on a design or performing an evaluation."

To this end they developed the "activity checklist" which we describe in the next section.

6 The Activity Checklist

6.1 Background

The activity checklist was presented in 1999 in a paper [13] published in the HCI journal *Interactions*. Its goal was to provide a more accessible formulation of key concepts in activity theory for application to software design and evaluation. The paper was authored by Kaptelinin and Nardi, two of the leading proponents of activity theory's application to HCI, and Macaulay who was acting as a practitioner.

The paper itself contains a testimonial by Macaulay, who explains in general terms why the checklist proved useful. But in the years since its widespread dissemination it has seen only limited use. Moreover, when the checklist has been used, it has been applied in various ways. Turner and Turner [18] used the checklist as the basis for semi-structured interviews, but gave no details on how exactly they derived their questions. Cluts [9] did the same, although used it to inform observation. Fjeld et al. [11] used the "sample questions" provided in the checklist paper to fuel their analysis. In contrast to Turner and Tuner, they answered the questions themselves rather than using them in interviews. They did not comment on how limiting their research to the example "sample questions" might have affected their research. Irestig et al. [12] used the checklist as a framework for discussing case studies of prototypes from usage centred and participatory design methodologies. They argued that the checklist might have biased their discussions away from issues surrounding the management of "signs and symbols", which they felt to be of great significance and not adequately addressed in the checklist.

6.2 Overview

The checklist consists of a number of points to consider when analysing an activity. There are two versions of the checklist which contain small variations tailored for use in either evaluation or design. The checklists contain a large number of points, 37 and 43 items respectively. Examples of an evaluation and a design checklist item are as follows:

> Eval 3.5: Use of target technology for simulating target actions before their actual implementation
>
> Design 3.3: Possibilities for simulating target actions before their actual implementation

The checklist was designed to provide a concrete version of the conceptual system at work in activity theory. This was specifically aimed at making activity theory "more useful" for evaluation and design of interfaces. At the same time they suggest "the Checklist orients without prescribing", which could reduce its usefulness in research contexts where a more prescriptive resource is desirable.

The checklist is divided into the four categories of *Means/ends*, Environment, *Learning/cognition/articulation*, and *development*, each of which are important higher-level concepts in activity theory.

The paper also includes a number of sample questions divided into the same categories, although not specifically tailored to evaluation or design. These questions are presumably derived from the checklist, and provide examples of analysis working above the level of individual checklist items.

The checklist paper contains a testimonial from Macaulay who used the checklist as an integral tool in her research, and the following quote illustrates one of the ways in which it can be used:

> "It gave me a quick way of relating experiences in the field to AT concepts. It helped me think about the kinds of data I wanted to gather, and the kinds of questions I wanted to ask." [13]

7 Checklist Critique

While Kaptelinin and Nardi claim that the checklist makes activity theory accessible, the reality may be quite different. Our colleague Brown [8][p.50] points to the disappointing state of the checklist's adoption, and suggests the following explanation for this:

> "For all items, the language used is taken from cultural-historical psychology, hence the checklist is of little use to those without a basic knowledge of the activity theory framework, and this may be a reason for its limited adoption."

However, the difficulties with the checklist go deeper than this. The following critique is based on our experience of using the checklist in a detailed study of electronic music producers and their tool use. We used the checklist as the basis for carefully developing a series of coherent interview questions. These questions were then used in interviews of five professional music producers. The interviews were semistructured, covering the questions in order, but following conversation threads as they arose naturally. Each interview lasted between one and four hours. In some cases these were then complemented with interviews using the cognitive dimensions framework [5,4], open ended discussion, and observations.

In an effort to examine the learnability of the checklist the lead author sat in on a post-graduate seminar discussing papers on activity theory and the activity checklist. The students experience with activity theory ranged from many years through to no prior exposure. For several students activity theory was a central component of their masters thesis research. Discussing the student's perceptions of the checklist in this context provided a useful touchstone to compare with our experience of understanding and using the checklist in our own research. The comments from this seminar are also reflected in the analysis below.

In the sections below we identify and explain issues which principally apply to HCI practitioners attempting to use the checklist as a gateway into activity theory, and who have limited previous exposure to activity theory concepts. We should emphasise that our aim here in identifying these issues is to work towards remedying specific weaknesses in the otherwise excellent approach and foundation we find in the activity checklist.

7.1 Operationalisation

The first issue with the checklist is that the items are not presented in an operationalised form. The checklist items are worded in a way which does not immediately lead the mind to apply the checklist item. The following item is typical wording from the checklist:

Eval 3.4: Self-monitoring and reflection through externalization

This phrase in itself does not directly pose a question, or impel the researcher to do anything. In order to apply this item, the practitioner must first comprehend the sentence, and convert it into a meaningful question or instruction. While it may not be difficult to do this for any single checklist item, having to interpret a list of more than forty such phrases suggests barriers to the comprehension and accessibility of the checklist. There are two problems this presents. HCI practitioners first encountering the checklist and activity theory may find it hard to understand how the many checklist items would actually be applied. Also, because the checklist is intended to be used as a quick reference guide in the field, for example during a semi-structured interview, obstacles to comprehension could make fluid discussion difficult. In an interview or other domain setting, HCI practitioners need the content of a checklist to be rapidly accessible so that they can quickly determine the relevance of various items to the current context.

7.2 Different application of points

Many of the checklist items directly address elements of the activity being examined, while others speak to a higher-level of analysis. The following item presumably calls the practitioner to look at issues around how goal conflicts are or could be resolved in the activity in question:

Eval 1.8: Resolution of conflicts between various goals

In contrast, other items seem to call for the practitioner to perform higher-level analysis of their findings, such as the following item:

Eval 1.9: Integration of individual target actions and other actions into higher-level actions

The potential differences in how these items can be applied may also add to difficulties in quickly understanding and applying the checklist.

7.3 Jargon

Another problem with the checklist is that while it attempts to make the ideas of activity theory "concrete", it uses activity theory jargon which negates this. In particular, the checklist uses specialised meanings of the terms *action* and *operation*, and *internalisation* and *externalisation*. The checklist preamble uses these terms but does not define them clearly.

The concepts and terminology are clearly explained in the accompanying article, but this does not provide an easy or quick reference to a practitioner learning with the checklist in the field. As an example, the following checklist item requires the practitioner to recall activity theory's technical definition of internalisation:

Design 3.1: Components of target actions that are to be internalized

7.4 Number of items

Depending on the version used, the checklist has between 37 and 43 items. In order to make the checklist easier to apply and internalise, it would be desirable to minimise the number of checklist items where any redundancy is present. A number of the items in the checklist closely relate to each other and could be explained as different specialisations of a single activity theory concept. For example, activity theory conceptualises actions and goals as being analysable in terms of sub- and higher-level paired goals and actions. The checklist includes three items that essentially prompt the practitioner to analyse this dual hierarchy. These items are distributed among other less related checklist items:

Design 1.2: Goals and subgoals of the target actions (target goals)

Design 1.4: Decomposition of target goals into subgoals

Design 1.9: Integration of individual target actions and other actions into higher-level actions

If these types of closely related items could be captured in a single item this might make the checklist smaller, more meaningful and easier to internalise.

7.5 Design vs. evaluation

Kaptelinin and Nardi's approach divides the possible usage of the checklist into two camps, evaluation or design. Providing two versions of the checklist creates a false dichotomy, making it unwieldy for use in the in-between cases. For example, HCI practitioners hoping to use the checklist to understand current tool use with the explicit goal of creating a new and *as yet undefined tool* may find they want to use elements from both checklist versions. However, it is far from obvious at first glance how and where the different checklists vary without detailed examination.

Other methods such as *contextual design* include the evaluation of current systems and work in the design of new systems:

> "Use contextual interviews to see *how people use the current system* or prototype, with a focus on how the system gets in their way or interferes with their work." [3] (Emphasis added)

The information collected from these sorts of *evaluation* interviews feeds directly into the design processes.

7.6 Readiness for application

Another potential concern with the checklist is that even though it is supposed to provide a concrete activity theory resource, there is some conceptual work needed before it can be practically applied. In particular, one suggested way to use the checklist is as a source of issues to cover in an interview. Unfortunately, the difficulties we have already discussed above are an obstacle to this.

As a first step to using the checklist in this way, HCI practitioners need to rework the checklist into a number of coherent questions presented in language that subjects in the domain of study will easily understand. Developing such a resource from the checklist requires an expertise and understanding of activity theory that many HCI practitioners may not have. This creates an unnecessary barrier to the checklist's adoption.

Activity theory argues that learning happens through the process of *internalisation* embedded in a real life activity system. Kaptelinin and Nardi suggest that ideally this process will be at work in the use of the checklist itself, where the HCI "practitioners should familiarize themselves with the Checklist and even try to internalize it". Not only will the HCI practitioners be internalising the checklist itself, more importantly they will be internalising the central activity theory concepts. This process is most likely to become "concrete" in the application of the theory to the real world domain that the practitioner is investigating, often through the use of interview and observation.

Instead of having to develop a series of interview questions based on a theory they have not fully internalised, HCI practitioners could make good use of a series of checklist based interview questions using everyday language. In applying these questions in their domain of investigation, a more concrete understanding of activity theory could be more easily internalised. This concrete understanding of the theory embedded in their own domain of interest would complement their readings on activity theory. In this way, HCI practitioners would begin to understand the meaning of the checklist through the nature of their subjects' responses, and their analysis would unfold naturally. In the next section we introduce a series of interview questions which we have designed to fulfil this purpose.

8 The Activity Interview

We have found there to be a clear need for a reference set of activity theory interview questions. This need is shown both by the checklist authors' comments, and by the difficulties raised above. This section introduces a new set of interview questions that we developed in order to meet this need.

Section 6.1 introduced some of the different attempts to apply the checklist in the analysis of activities. One approach seen in some of this prior work is the questioning of subjects directly to obtain their personal activity theory analysis of their activities. The traditional empirical observation model would cast the observer as the impartial and analytical expert, using their theoretical knowledge to dissect the situation. Modern understanding of qualitative methods has since moved the emphasis to the subjective nature of such analysis. This has typically resulted in two approaches. Either researchers emphasise the importance of their own analytical bias in their research, or they attempt to provide minimally mediated accounts of subjects' own understanding.

There are advantages and disadvantages in applying the extreme form of either subjective approach. On the one hand, relying on the external analysis of only the researcher neglects the deep insight that subjects' have about their own activities and behaviour. However, it does allow the researcher to choose an analytical "lens", or framework, appropriate to their goals. On the other hand, uncovering subjects' understanding of their own activities can expose important facts about how their own conceptions are organised. Unfortunately this can lead to accounts that may be of little pragmatic use given that they can overemphasise the concerns of the subject rather than what the observer may be trying to address.

In using a set of interview questions which implicitly contain the conceptual framework of activity theory we can gain some of the advantages of both approaches. A similar method can be seen in the cognitive dimensions questionnaire [6,4], a technique for evaluating notational systems which depends on the critical insight of a tool's expert users. Presenting subjects directly with the concepts of analysis allows them to express their own deep understanding of their activity, but with the language and focus that will be useful for the practitioner. This technique places a value on the ability and insight of the subject which is in line with the philosophy embedded in activity theory. Activity theory takes the position that humans are highly capable of extending and developing themselves through their use of analytic or physical tools. However, we will also see how the role of the practitioner's external analysis has a place in this, and how it meets the implicit intention in the design of the checklist.

In the sections below we outline how we developed the activity interview and give some examples of the questions derived from the checklist.

8.1 Design goals

A number of requirements for the interview resource can be identified by explicitly addressing the weak points of the checklist that we identified in the critique in section 7.

Of course the principle demand for interview questions is that they do not contain jargon. Interview questions will be of very limited use if they contain unusual terms, or words with special technical meanings. Wherever a word with a special technical definition is used, it is important that the context provided by the question makes such usage clear.

By removing activity theory jargon from the interview the practitioner may loose sight of the relation of specific questions to the theory's concepts. The interview resource should contain non-verbalised references to activity theory concepts and terminology at appropriate points on the interview sheet. This will provide both a means for HCI practitioners to internalise these concepts in the context of their domain, as well as helping them relate the subjects' responses back to activity theory proper.

In order to gain the leverage of the checklist, we need to ensure our questions provide full coverage over all of the checklist items. If the benefit of the checklist is to give HCI practitioners quick access to the important activity theory concepts, these all need to be included in some manner in the interview.

Full coverage of the checklist could lead to excessively long interviews. For this reason it makes sense to attempt to minimise the number of questions through amalgamation where appropriate. Many of the checklist items relate very closely together, and can be seen as different aspects of the same concern. Addressing these in a unified manner in a single question should lead to more coherent, and more informative responses from subjects.

Finally, ideally the interview questions should be able to be used in a wide range of contexts. The evaluation and design dichotomy present in the checklist can prove to be a barrier where the practitioner is interested in investigating the role of current tools for the design of a new, as yet undefined tool.

8.2 Method

The general method for creating the interview questions consisted of a number of steps. Firstly it was necessary to systematically merge the evaluation and design versions of the checklist. It is non-trivial to identify the differences between the two versions by simply reading them. Often Kaptelinin and Nardi have placed the identical item in both checklists. In some cases the item is reworded specifically for evaluation or design. Some additional items are only included in one of the two versions.

In order to reduce the number of checklist items and increase the coherence of interview questions it was necessary to find items which were near duplicates, overlaps or closely related to each other.

The next step was to reword the resulting items in everyday language that carried with it important activity theory concepts. The resulting set of questions were then checked for coverage over the original checklist items and that they were ready to be used in interviews.

Finally, we used the checklist in our own research of computer music production, and used the experience to refine the interview questions for future use.

8.3 The questions

In this section we introduce a number of the questions we have developed for the activity interview, and explain their origins in the activity checklist. While the full set of activity interview questions are not included here, this discussion serves to demonstrate the way in which they were developed. The complete activity interview resource can be found at the end of this paper.

We applied the method detailed in the sections above to the two versions of the checklist, but excluded the reflexive items that looked at the design activity itself. The "evaluation" version contains 37 items, and the "use" component of the design version contains 34. After carefully amalgamating items which were slight rewordings from the different versions of the checklist we were left with 41 unique items. Our work outlined here resulted in 32 unique interview questions. Below we discuss some of the more interesting cases.

Several groups of original checklist items were combined for the final question list. Activity theory's highly related notions of the Activity, Action, Operation; and Motive, Goal, Condition hierarchies are addressed by several items. The following items all call on the practitioner to determine the elements of this dual hierarchy in their domain:

Design 1.2: Goals and subgoals of the target actions (target goals)

Design 1.4: Decomposition of target goals into subgoals

Design 1.9: Integration of individual target actions and other actions into higher-level actions

The distinction being made between the first and second item is far from clear. Additionally, since goals and actions are so closely related it does not seem ideal for them to be addressed in separate interview questions as seen in the third item. Instead, we can ask subjects to reflect on how they achieve their tasks, and how these actions relate to their goals. This allows the third item to be integrated with the other two into the following compound question:

Interview 1.2: *What are your ultimate goals in your role, and how do you achieve them? What are your goals along the way*?

By presenting several closely related checklist items as a single compound question, we can expect the subject to reflect on the important relationships between their goals and sub-goals, and the actions that they use to achieve them. Addressing these issues as separate questions could lead to a fragmented and incomplete response.

Another example of this type of amalgamation is our approach to several checklist items that deal with externalisation:

Eval 3.8: Coordination of individual and group activities through externalization

Eval 3.9: Use of shared representation to support collaborative work

Eval 3.10: Individual contributions to shared resources of group or organisation

Each of these items overlaps the others, but it not immediately obvious exactly how. The first item deals with both individual and group activity, which means all activity. It emphasises coordination through externalisation. The second item refers to shared representation, which is a form of externalisation, and focuses on just group activity. The first item's issue of coordination through externalisation is simply a special case of the second item – a shared representation to support collaborative work. The third item deals with contribution to shared resources, which are another product of externalisation. As all contributions to groups can be traced back to individuals, it can instead be described as a case of group activity. All of these items can be essentially captured by an umbrella question uncovering what is externalised, and how externalisations are used – including including use for coordination. After removing the jargon, we can express this to subjects in the following question:

Interview 3.4: *How do you use representations of your work – documents, notes, software, and talking etc. – to collaborate and coordinate with others?*

To use the checklist as a unified tool to support evaluation for prospective design, it was necessary to discover the variations in wording between the evaluation and design versions of the checklist. Often the differences in wording do not make a significant change to the item, and simply create potential confusion for HCI practitioners wanting to utilise elements of evaluation and design. For example, the following two items are based on the same underlying activity theory concept:

Eval 3.3: Time and effort necessary to master new operations

Design 3.2: Time and effort necessary to learn how to use existing technology

Both of these items are attempts to determine how subjects learn to use tools in a specific activity. The same items can both be addressed in an interview context with the following question:

Interview 3.2: *How hard did you find it to master your tools, and what should have been easier?*

There are several checklist items that include activity theory jargon which would not be understood by interview subjects. One such example that was discussed above is the use of the term "externalisation". Another example is contained in the following item which uses the technical terms "actions" and "operations":

Eval 1.13: Support of mutual transformations between actions and operations

Again, it is non-trivial to reword this item as an interview question that can be easily understood. HCI practitioners who are learning activity theory may not be ideally placed to try and frame this in the clearest terms for their subjects. We could address this by asking:

Interview 3.6: *How do your tools support the transition between subconscious and conscious use?*

The above interview questions are just a selection of our full set of 32 questions found at the end of this paper. It should be taken into account that these questions are still a work in progress, although they did prove useful to us in in our own research. In the next section we will turn our attention to this experience and comment on how well the interview questions worked in action.

9 Activity Interview in Action

In this section we briefly look at how we used the activity interview in our own design process, and then reflect on the success of the interview itself. We used the new activity interview as one of several techniques in our study of computer mediated music production. The other methods included interviews with the cognitive dimensions questionnaire, open discussion, and observations. The activity theory interview component was conducted with five subjects, including professional music producers and a physical music interface designer. Each of these interviews took between one and four hours, depending on the depth of the subject's answers and the time they had available.

In general, this new activity interview tool worked well. Subjects gave meaningful and often insightful answers that related directly to corresponding activity theory concepts. This helped minimise two forms of researcher bias. Firstly, the questions are based on activity theory without being tailored by the researcher to the specific domain being studied. This means there is more leeway for the participants to draw unforeseen connections between their domain and activity theory concepts. Therefore the researchers expectations do not unduly limit the results. Secondly, basing analysis solely on observation and resulting questions would be inherently limited. Producing a completed piece of music will typically stretch from weeks to years, a time scale that makes detailed observation of the entire process impractical. Obviously, the subjects themselves are the only people with knowledge of their entire activity, and how it unfolds over these macro-timescales. This form of bias stems from the researchers limited exposure to the entirety of the activity, and can only be offset by incorporating the subject's own reflection. With the addition of observation and the researcher's analysis we can hope to get closer to a full picture of the activity.

9.1 Process and results

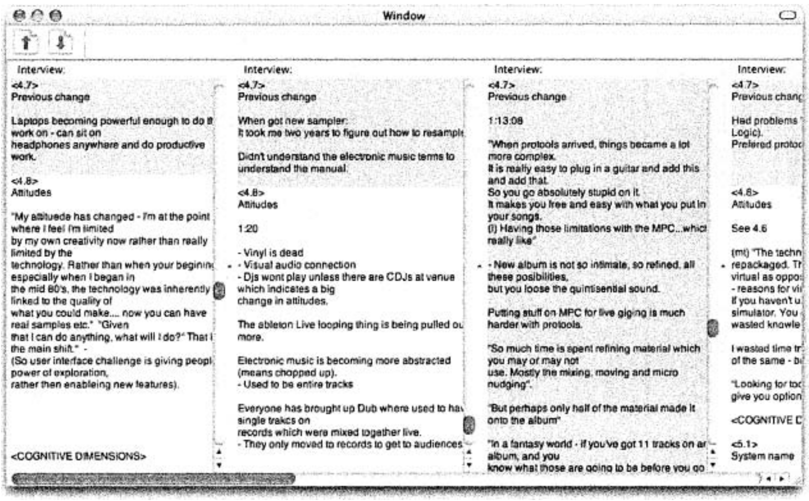


Fig. 2. A software tool we developed to help with analysing activity interview responses.

Practitioners should not expect any trivial one-to-one mapping between an activity theoretic analysis, and their system design. However, such an analysis is a crucial foundation for successful design. In our design process we found that our activity interview helped us ensure that we had covered all of the major issues in understanding the activity system with each interview subject. In order to work through the interview material, we made detailed notes from our audio recordings, and marked them up with simple text tags to organise them by question number.

We then used a software tool we developed to display and highlight all of our interview participants' responses for a given question. When we have a design question addressed by a specific area of activity theory, we can instantly view the responses from the relevant interview question. This tool can be seen in action in figure 2.

The information that we uncovered from these interviews was invaluable and of course is having a direct impact on many aspects of our still unfolding design concepts and prototypes. For example, by highlighting the *activity* as the unit of analysis, rather than just lower-level tasks, the interviews brought out the false dichotomy created in many tools between the studio and live performance. Producers' compositions are typically being crafted for dual use – both for distribution on audio media such as CDs, but also for performance in a live context. Additionally, producers often expressed that their work was never truly completed, and that album, single, and re-mix versions of songs were common. Therefore a design goal of our system has become to allow producers to manage multiple variations of a composition, and allow modifications of one to naturally be reflected in the others where appropriate.

The activity interview also brings out issues of collaboration. For example, we found that often more than one person would want to interact with the computer representation of the composition. This raises interesting design possibilities of having multiple people working on different aspects of a shared composition simultaneously. With collaborators often at times working on developing a composition apart, it is also important to facilitate merging diverging versions.

The activity interview has had far too many additional impacts on our unfolding design to describe here, but the above examples are indicative of their character.

9.2 Higher-level questions

In addition to the checklist items, the checklist paper also provides 23 "sample questions". The sample questions demonstrate the type of questions that the HCI practitioner could ask about an activity. An example of one of the provided questions is:

> Sample 3.4: Does the system provide representations of user's activities that can help in goal setting and self-evaluation?

This question combines checklist items concerning externalisation *and* goals. The sample questions were compelling enough for both Macaulay [13] and Fjeld et al. [11] to use explicitly in their research. Macaulay described them as being "particularly useful.". However, these are sample questions that are representative of the *type* of questions that should be asked, but far from a complete resource of the questions that should be considered. By using the sample questions as a primary resource there is a very real danger of failing to consider other important questions not included in the samples. The underlying problem here is that the checklist paper provides no guide on how to derive similar questions. In fact the main body of the checklist paper does not mention them at all.

On closer examination, rather than being merely the checklist items reworded as questions, the sample questions work at a higher level – each synthesising several orthogonal checklist items. The real value of the sample questions lies both in their operational nature (as opposed to the abstract checklist items), and in how they make links between activity theory concerns. The unfortunate consequence of this is that attempting to create an exhaustive list of these higher-level questions would result in a combinatorial explosion which would be far from practical.

Our newly developed activity interview addresses this through use in a semistructured interview context. This takes advantage of the natural ability of people to make linkages between comments over the duration of a conversation. As these linkages naturally occur to the interviewer or subject it is important to follow these connections with followup questions. Such spontaneous questions from the interviewer, and insights from the subject are a natural way to discover relevant replacements for the sample questions.

One of the strengths of our newly developed activity interview is that it provides the vehicle for maximising the number of such connections that will be made. By raising all of the salient checklist items in discussion with subjects, these higher level questions naturally arise. Without such a set of interview questions there is a real danger that the researcher will be limited to only the provided sample questions.

9.3 Hierarchical goal analysis

Our initial activity interview included a question asking about goals and subgoals in the activity.

Interview 1.2: *What are the goals, subgoals and super goals you are trying to achieve in your role?*

One problem with this question was that it proved to be difficult to get a simple hierarchical account of actions and goals in music production. On reflection it became obvious that this is because there is no simple hierarchical description of music production. As introduced in section 3, this activity is open-ended and non-prescriptive, and as such the actual configuration of actions and goals varies between one musical piece and another, and between each subject depending on their patterns of collaboration, tools, and work habits.

Subjects had some difficulty in knowing how to answer this question. Subjects tended to answer in terms of a "for example" account of the actions they might go through in producing a track. They made it very clear that such an account was merely one possible scenario, and that in actual practice the order and details of their actions varied greatly. As the subject gave their account it was possible to query them about their goals along the way. This example shows the impracticality of asking subjects to develop a high-level and idealised abstract description of their actions and goals. They proved to be much more comfortable giving a concrete account, and providing cues as to how this can vary from case to case.

This interview question needs to be reformulated in concrete terms, with room for the subject to explain how the activity can vary:

Interview 1.2 *Can you take me step-by-step through the process of how you complete your activity, and tell me how this process can vary.*

<Ask about sub-goals during the account where appropriate>

9.4 Conflict

The checklist uses the term "conflict" in checklist items dealing with tensions between various goals or various actions. The paper itself does not explain this term in any more detail. The use of this term here is somewhat confusing. Activity theory literature has typically used the concept of *breakdowns* or the wider notion of *contradictions* as popularised by Engeström [10]. Even the book *Context and Consciousness* [16] which was edited by Nardi herself does not include "conflict" in its index.

Despite the unusual choice of this term, we decided to use it in the activity interview. One of the goals of the activity interview was to embody the checklist and the expertise it embodied. Changing the terminology unnecessarily would not be ideal. Unfortunately this term caused much confusion when used in interview format. One of the questions was:

Interview 1.5 *What conflicts are there between these goals?*

The subjects almost universally interpreted the term conflict to mean interpersonal conflict, rather than the more abstract idea a form of tension or contradiction between the goals themselves. It is interesting to note that the checklist had already limited its analysis of conflict to goal and action conflict, only a small subset of the types of contractions that can occur between parts of an activity system. For example, contradictions can occur at different levels between tools, roles, and rules in addition to goals and actions. Inadvertently limiting the analysis to just *interpersonal* conflicts over specialised it even further.

This also leads one to wonder how many HCI practitioners new to activity theory may be misinterpreting the meaning of the term "conflict" in this context. With no explanation in the paper, and a very real possibility of confusion, the use of this term is highly problematic. In the latest version of the activity interview we instead use the term "contradiction".

9.5 Internalisation

The checklist paper claims to address the problem that concepts such as internalisation and externalisation are " somewhat abstract when it comes to the actual business of working on a design or performing an evaluation". However, the checklist simply reuses these same terms but in an itemised form, which can cause difficulties. For example, in developing the first version of the activity interview we made the mistake of equating internalisation with the very different concept of a task's transformation to sub-conscious operations. Both internalisation and sub-conscious operations are important concepts in activity theory, but have very distinct meanings. This confusion reduced the effectiveness of the interviews, and as a result hindered the perceived usefulness of activity theory.

Issues regarding internalisation are highly important in our particular domain. In our other complementary interviews using the cognitive dimensions questionnaire, subjects made statements about how they had been "colonised" by the model and work procedures of their music production software. This phenomenon should have been explored in a rich and deep manner by any interview claiming to elicit an activity theory analysis. Our initial interview's flaws stemmed from the ambiguous and jargonised activity checklist. The latest revision of our activity interview has been adapted to remedy these limitations.

9.6 Externalisation

Externalisation is another central concept of activity theory. We found that our activity interview questions about externalisation did not get the detailed answers that we might have expected. Interestingly, the domain of music production is permeated by profound externalisation. Most of the subject's time is spent working with externalised abstract representations of the object of the activity. It is likely that activity in other domains which are more internalised would see an larger benefit from these questions in the interview.

9.7 Evaluation vs. design

Our activity interview merged the evaluation and design checklists. This suited our goal of analysing the *current* activity and tools, while looking forward towards how we would design a new tool.

In our interview's section on change and development of the activity, we asked about what might be possible in a new tool. This was followed with additional questions on how this tool could change the subject's activity. In this case, our subjects found it very difficult to imagine new tools, which we can attribute to their accounts of having been "colonised" by their current tools. Given the difficulty of imagining new tools, further questioning about the impacts of such hypothetical tools gained us little further insight. It is difficult to know how much we can attribute this phenomena to our specific domain, or if there is a more general problem here. One confounding factor is that producers who extensively use computer music tools for composition have by definition found a fit between their activity and their tools. Other potential computer music producers have no doubt been put off by the complexity of the tools, in addition to the patience and tenacity required to complete projects with them.

Presumably in many other domains there will be a well articulated need for new tools, and subjects will have a clear idea of what such a tool might comprise.

9.8 Limitations of the interview

With the interviews completed we can think about which aspects of the activity they failed to illuminate. The above sections highlight problems due to various factors, but there are some things that the interview simply omitted. Some of these limitations can be traced back to equivalent gaps in the checklist, or activity theory itself.

An important aspect of computer music production is how the design of tools affect the resulting musical outcome. We can expect similar concerns to exist in many other domains where we might want to use activity theoretic analysis. We found that the activity interview did not directly lead to discussion in this area. The closest the activity checklist gets to raising this aspect of activity is with the following item:

> Eval 4.2: Effect of implementation of target technology on the structure of target actions

However, the structure of actions is only a small part of the concerns here. It is of course important to develop an understanding of how tools affect the structure of our actions. We also need to know how the tool interacts with and shapes our goals, and the object of our goals. Tools both empower and constrain our abilities through their specialisation. When using a tool "we can sense some things better, and we can alter some things better, but others not at all" [15]. This needs a stronger emphasis in both the activity checklist and has been addressed in our interview.

As discussed in section 6.1 Irestig et al. [12] found that the activity checklist "does not highlight the management of signs and symbols in design as a significant and separate aspect of the process". We also foresaw this limitation, and dealt with it by also interviewed subjects with the cognitive dimensions questionnaire which focuses on important notational and symbolic features of tools. The detailed and crucial information uncovered in these interviews was not duplicated in the activity interview. This validates the claim that this is a limitation of the activity checklist, but we should not conclude that the checklist, or our activity interview, should be expanded to include these aspects. The cognitive dimensions questionnaire worked well in tandem with our activity interview. Finding research methods that complement this weakness seems the best course of action.

10 Conclusion

In this paper we have presented a new interview resource for informing user-interface design. This resource should prove useful to HCI practitioners seeking to understand open-ended and non-prescriptive activites. It consists of a number of interview questions derived from the activity checklist, which is based on cultural historical activity theory. We showed that while the checklist provides an excellent foundation, it has some unfortunate shortcomings for use as a design tool. HCI practitioners who are not steeped in activity theory literature may find it inaccessible and difficult to apply without significant conceptual work. While the checklist will continue to be useful in many research contexts, our interview resource is a pragmatic, accessible, and instantly applyable alternative, and aims to correct some of the checklist's limitations.

Future work includes the further refinement of the activity interview. This needs to be informed by both the outcomes of our current research, but also from the results of using the activity interview in other complex domains outside of computer mediated music production.

THE ACTIVITY INTERVIEW

<Discuss and define what the activity is for this person and resolve any questions>

GOALS

1.1: What are the different roles of the people involved in this activity? What is your role?
1.2: Can you take me step-by-step through the process of how you complete your activity, and tell me how this process can vary.
<Ask about sub-goals during the account where appropriate>
1.3: How can you tell when you have successfully completed each goal?
1.4: How could the higher-level goals be achieved differently?

CONTRADICTIONS

1.5: What contradictions are there between these goals?
1.6: What contradictions might there be with goals from other activities you or others are involved in?
1.7: How could you / do you resolve these contradictions?

TOOL USE

I will now ask you a series of questions about the tools you use in this activity. Tools include all sorts of artefacts such as pencils and paper, the notes we take, the theories behind them, and more traditional technology tools.
<Give other examples from the specific activity domain>
2.1: What tools do you use and what for?
2.2: What other tools do you or could you use in this activity?
2.3: Do you have access to these tools?
2.4: How could the tool be integrated with your other tools?
2.5: Which tools do you use when, and how do you organise them?

WORKING WITH OTHERS

2.6: How are these tools used and shared with others?
2.7: How do you divide this activity between people, and at which points do you have to wait for them to complete their work?
2.8: What are the explicit or implicit rules, norms and procedures influencing how you work?

INTERNALISE

3.1: How have your tools affected how you think and reason about your activity, and how much of your activity do you perform in your head?
3.2: How hard did you find it to master your tools, and what should have been easier?

EXTERNALISE

3.3: How do you deal with problems in this activity when they become too complex to manage in your head?
3.4: How do you use representations of your work – documents, notes, software, and talking etc. – to collaborate and coordinate with others?
3.5: How do you externalise processes for simulation?

TRANSITION

3.6: How do your tools support the transition between subconscious and conscious use?

HELP

3.7: When things go wrong, how do / could your tools help you express these problems and request help?
3.8: How does the system provide help to other people?
3.9: What knowledge is there about these tools (other than that provided by the tools), and how can you get access to it?

LIFE CYCLE

4.1: How do your tools fit into your workflow?

4.2: How do your tools shape how you work, and what you can and do produce?

CHANGE

4.3: What new things could be possible in a new tool?
4.4: How will contradictions change between target actions and higher-level goals?
4.5: How might your working environment change - people, technology, rules, work etc. as a result of new tools?
4.6: What could you do differently with better support from a new system?
4.7: How have previous systems you have used affected how you performed your activity? When you moved from older to newer tools in the past, how did this affect the activity?

ATTITUDES

4.8: What are your attitudes towards new technology, and how do you see them changing over time?

References

1. Ableton. Live: The audio sequencing instrument, 2003. http://www.ableton.com/index.php?main=live.

2. L. Bannon, and S. Bødker, Designing Interaction: Psychology at the Human Computer Interface. Cambridge University Press, Cambridge, 1991, ch. Beyond the interface: Encountering artifacts in use.

3. H. Beyer, and K. Holtzblatt, Contextual Design : Defining Customer-Centered Systems. Morgan Kaufmann Publishers, 1998.

4. A. Blackwell, and T. Green, A cognitive dimensions questionnaire, 2000. http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/CDquestionnaire.pdf.

5. A. Blackwell, and T. Green, Notational systems - the cognitive dimensions of notations framework. In HCI Models, Theories and Frameworks: Toward a Multidisciplinary Science, J. M. Carroll, Ed. Morgan Kaufmann, 2003.

6. A. Blackwell, and T. Green, A cognitive dimensions questionnaire optimised for users. In Proceedings of the Twelfth Annual Meeting of the Psychology of Programming Interest Group (2000), A. Blackwell and E. Bilotta, Eds., pp. 137–152.

7. S. Bødker, Through the Interface. Lawrence Erlbaum Associates, 1991.

8. J. Brown, Activity theory: Its use and potential in the field of human-computer interaction. PhD Comprehensive, Psychology Dept, Carleton University, Ottawa, Canada, July 2004.

9. M.M. Cluts, The evolution of artifacts in cooperative work: Constructing meaning through activity. In GROUP '03: Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work (New York, NY, USA, 2003), ACM Press, pp. 144–152.

10. Y. Engeström, Activity theory as a framework for analyzing and redesigning work. Ergonomics 43, 7 (July 2000), 960–74.

11. M. Fjeld, M. Morf, and H. Krueger, Activity theory and the practice of design: evaluation of a collaborative tangible user interface. International Journal of Human Resources Development and Management 4, 1 (2004), 94–116.

12. M. Irestig, H. Eriksson, and T. Timpka , The impact of participation in information system design: a comparison of contextual placements. In PDC 04: Proceedings of the Eighth Conference on Participatory Design (New York, NY, USA, 2004), ACM Press, pp. 102–111.

13. V. Kaptelinin, B. A. Nardi, and C. Macaulay , Methods & tools: The activity checklist: a tool for representing the "space" of context. Interactions 6, 4 (1999), 27–39.

14. A. Leont'ev, Activity, Consciousness and Personality. Prentice Hall, Englewood Cliffs, NJ, 1978.

15. M. Mc Cullough, Abstracting Craft: The Practiced Digital Hand. The MIT Press, London, England, 1996.

16. B.A. Nardi, Ed. Context and Consciousness. MIT Press, Cambridge, Massachusetts, 1996.

17. D.A. Norman, The Invisible Computer. The MIT Press, 1998.

18. P. Turner, and S. Turner, A web of contradictions. Interacting with Computers 1, 14 (2001).

19. L. Vygotskii, Mind in Society : The Development of Higher Psychological Processes. Harvard University Press, 1978.

Visual Representation of Complex Information Structures in High Volume Manufacturing

Connor Upton and Gavin Doherty

Distributed Systems Group, Department of Computer Science, Trinity College Dublin
{connor.upton, gavin.doherty}@cs.tcd.ie

Abstract. While research supports the use of graphic data representations in interfaces and control systems, work in this area has focused on relatively small systems with a limited number of variables. This paper describes an approach to designing a visual application for a semiconductor manufacturing plant. This is a complex, large-scale system requiring a structured design methodology. First, using cognitive work analysis techniques an Abstraction Decomposition Space (ADS) of the system is generated. Second, as with ecological interface design, we demonstrate how this ADS can inform the display design. The complexity and scale of the system has required us to make adjustments to both of these frameworks. The resulting display req multiple views of the system, information hiding and user interaction. Tak wider set of analyses onboard, we present a design rationale supportin explicit representation of hierarchies, the compatibility of views and the u contextual navigation.

1 Introduction

The visual representation of data in control interfaces has been shown to increase performance and reduce human error [1, 2]. Research into the area of external cognition has indicated that the correct graphical encoding of data allows us to exchange cognitive operations for perceptual operations thus improving performance [3]. Techniques for the encoding of data using visual variables have been explained and validated [4, 5]. While the knowledge gained from this work can be used to inform visual design, much of the experimental work has focused on small problem spaces involving a limited number of variables [3, 6]. In these situations improved performance can be attributed to the externalisation of low-level cognitive operations such as search, comparison and integration, freeing up short term memory to focus on higher level tasks.

For interactive systems a structured approach to design is required. Data representations should not be created in isolation but must be considered in relation to the overall system. Various different approaches to user studies, requirements gathering and design evaluation already exist. While these are helpful in defining information requirements and useful for post-hoc design evaluations, there is relatively little information on how to transform information requirements into visual displays. This has been referred to as "the Design Gap" [7]. The design gap can be attributed to the large number of factors that effect visual displays and the diverse areas of research which are relevant to this field. In this paper we present a complex

Please use the following format when citing this chapter:

Upton, C., Doherty, G., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 27–45.

socio-technical system and outline an approach for revealing user requirements. We then present a design rationale that creates a visual display informed by our initial system analysis coupled with a range of design guidelines, thus bridging the design gap for this particular problem.

1.1 Complex Sociotechnical Systems

Complex sociotechnical systems are work environments that involve large problem spaces, multiple users & high-levels of automation. As open systems they frequently feature conflicting constraints, dynamic data, coupled components and unanticipated events. Examples of such environments include industrial process control, air traffic control and surgery theatres. Workers in these environments observe and interact with large volumes of real-time system data. This data is often multivariate with complex relationships existing between data sets. The cognitive activities involved in such systems go far beyond the low-level operations mentioned above. The complexity and scale of these systems means that data must be structured in a manner that is meaningful to end users before it can be visualised.

1.2 Design Framework

Ecological Interface Design (EID) [8] is a framework for designing interfaces for complex systems. It takes a two step approach first specifying the content to be displayed and then designing the visual form. While the framework is well defined it can be extended in a number of ways.

Firstly, the framework was originally defined in relation to process control. In order to test the frameworks generalisability it is important to apply it to different domains. Here we apply it to a High Volume Manufacturing environment.

Secondly, the example in the original framework deals with a representational microworld. A microworld is a pared down version of a real world system. It maintains aspects of a systems complexity while being simple enough to carry out accurate controlled experiments. The DuressII [8, 9, 10] system, around which the EID framework was developed, benefited from having a relatively small set of variables, a simple physical structure and a single operator. Our study deals with the large scale, highly complex process flow involved in semiconductor manufacturing.

Thirdly, while the principles of visual design proposed by the framework provide us with guidelines, they do not deal with the actual visual rendering of the components. This allows the framework to remain general. With smaller domains the principles can be achieved through rudimentary graphic techniques such as spatial arrangement. However large scale systems require more sophisticated representational techniques. We discuss the application of these techniques to the High Volume Manufacturing domain.

2 The EID Framework

Here the two stages involved in the EID framework are explained. The first stage involves the specification of content through the generation of an Abstraction Decomposition Space (ADS) [9] a tool used in cognitive work analysis. The second stage applies three principles of visual design to the content in the ADS to inform the interface design.

2.1 Specification of Content

Cognitive Work Analysis (CWA) [10] is a methodology created to analyse complex socio-technical systems. It takes a different analytical approach to other Human Computer Interaction research methods. Instead of looking at a work domain in terms of users and specific contexts of use; it aims to describe the complete domain in terms of the systems constraints. This generates a field description of the entire system. A good analogy for explaining field descriptions is that of helping someone to find their way to a location. User focused research will describe how a task is completed by an individual. It specifies a number of actions carried out to complete the task. It can be compared to giving someone a set of verbal instructions on how to get to a location, take the first right, then the second left etc. A field description describes the constraints under a task can be completed. It is more comparable to handing someone a map with their current location and target marked. This allows them to find their way but also to adjust their route should the need arise. A field description accommodates different worker roles in the same system. It can deal with non-normative work scenarios and can provide a more accurate model of the system.

Aggregation - Decomposition

Physical - Functional	Whole System (DURESS)	Subsystems (Reservoir)	Components
Functional Purpose	Outputs to Environment		
Abstract Function		Mass/Energy Topology	
Generalised Function		Liquid Flow & Heat Transfer	Liquid Flow & Heat Transfer
Physical Function			Component States
Physical Form			Appearance & Location

Fig. 1. The Abstraction Decomposition Space for DuressII [10]

The Abstraction Decomposition Space (ADS) [11, 12] is an analytical tool used in CWA to create a field description of the work domain by combining a decomposition hierarchy with an abstraction hierarchy. A decomposition or part-whole hierarchy splits a system into its subsystems and then subsystems into components. It reduces complexity by dividing a system into smaller units. An abstraction hierarchy is a description of a system in terms of functionality, from high-level goals down to the physical description of individual components that carry out basic physical tasks. Rasmussen proposes five divisions for process control; functional purpose, abstract function, general function, physical function and physical form. The ADS places these hierarchies orthogonally against each other providing us with a multilevel view of the system where each level describes the entire system at a different granularity. DuressII was a thermal-hydraulic process simulation around which the EID framework was developed. Figure 1 shows the ADS for this system. The ADS can be used to define the information requirements of individual users through use-case mappings [10].

2.2 Design of Visual Form

The EID framework outlines three principles for visual design of interfaces based on the Skills, Rules, Knowledge (SRK) taxonomy [13]. This taxonomy defines three levels of cognitive control that a user can exert over a system, distinguished by the manner in which the system is represented internally. Skills Based Behavior (SBB) involves reactive behavior to real-time system data. It describes how an expert user responds to temporal information about system components to maintain stability. Generally these actions are so fluid that they become instinctive and are not verbalised by users. Rule Based Behavior (RBB) relates to procedural tasks that follow a plan of action. They are guided by rules defined by the constraints of the system. While they can be learned and practiced to the point of fluency the actions can generally be recognised and verbalised by the user. Knowledge Based Behavior (KBB) relates to higher-level decision making. It generally involves reasoning at multiple levels of abstraction and requires knowledge of the complete relational structures in the system. KBB is used in fault diagnosis and performance analysis.

2.2.1 Levels of Cognitive Control

The way in which information is interpreted facilitates different levels of cognitive control. Vicente provides a good example with the controlling of a valve using a flow-meter [10]. The flow-meter consists of a measurement scale, a target indicator and a pointer. Given the task of stabilising the flow at the target the flow-meter becomes a signal. The time and space data given through the flow-meter gives the operator temporal feedback facilitating control. This is Skills Based Behavior. If the user closes the valve but the meter still indicates a flow, it becomes a sign. The mapping between the valve and the feedback from the flow-meter indicates that there is a mis-calibration between the meter and the valve. This is Rule Based Behavior. If the user recalibrates and the problem arises again the meter becomes a symbol. It becomes an element in the overall system that is related to other components and through it we can diagnose a leak in the valve or another fault in the system. This is an example of Knowledge Based Behavior.

2.2.2 Principles of Visual Design

The flow-meter example describes cognitive control of a single component through a physical interface. With modern sensor and display technology it is possible to integrate the control and feedback display for entire systems into a single digital interface. The EID framework notes that careful consideration must be given to the representation of system data to ensure safety and efficiency of use. Three principles of visual design are provided, each associated with supporting a level of cognitive control.

1. SBB – to support temporal control of a system direct manipulation should be used. Also the representation should be isomorphic to the part-whole structure.
2. RBB – provide a consistent one-to-one mapping between constraints and the cues or signs provided by the interface.
3. KBB – represent the work domain in the form of an abstraction hierarchy to serve as an externalised system model.

2.3 Application of EID

As a representative microworld DuressII sought to capture many of the characteristics of process-control in complex systems, while remaining simple enough to facilitate experiments. To achieve this, the complexity of the system was limited. The presence of a dual supply system introduced the complexity associated with coupled components. Despite this coupling, the overall process can still be identified as having a linear supply & demand relationship. The scale of the system was also limited. A small number of subsystems and components gave a total of thirty seven system variables. Finally, the system was designed for a single operator.

The limited scale and complexity of the system made it easier to follow the design principles set out by EID. SBB was satisfied by providing visual representations of the component related data and their associated controls on-screen. RBB was satisfied by showing all components in a single view and by making their coupling visually explicit. This would not have been possible with a larger system or more complex coupling. KBB was supported by grouping components according to their subsystem. This allowed the ADS to be represented by way of visual chunking.

In the remainder of this paper we apply the EID approach to the domain of High Volume Manufacturing. We note a number of differences between the two systems and make adjustments to the EID approach in order to accommodate them.

3 Case Study: Semiconductor Manufacturing

Modern High Volume Manufacturing (HVM) environments are examples of extremely complex socio-technical systems. They combine sophisticated factory automation with the changing demands of dynamic markets. A common constraint across HVM is the conflicting goals of achieving high volumes of production while ensuring that equipment continues to operate within acceptable quality control limits. High production volumes place equipment under stress, which may affect the quality of the product and the overall product yield. This requires equipment to receive more maintenance and repair. However, repair causes more downtime leading to lower levels of production. This conflict is generally resolved by humans who must reconcile manufacturing (production) focused and engineering (quality) focused priorities.

3.1 Description of System

Semiconductor Fabrication Plants (Fabs) are HVM environments involving hundreds of machines (described as tools) and a highly complex process-flow. The overall production process between raw silicon and final semiconductor is divided into a number of segments. Segments can be further subdivided into a number of functional operations that build parts of the semiconductor device. As semiconductor manufacturing is a multilayered process, some operations may be repeated in different segments, introducing circulation and re-entries into the process-flow. Operations are carried out on specific tools which are categorised according to specific functional activities; for example etching or lithography. Multiple tools carrying out the same operation are gathered together into a toolset. Groups of toolsets that carry out the same general function form a functional area. This complex relationship between process-flow and functional areas is shown in fig.2.

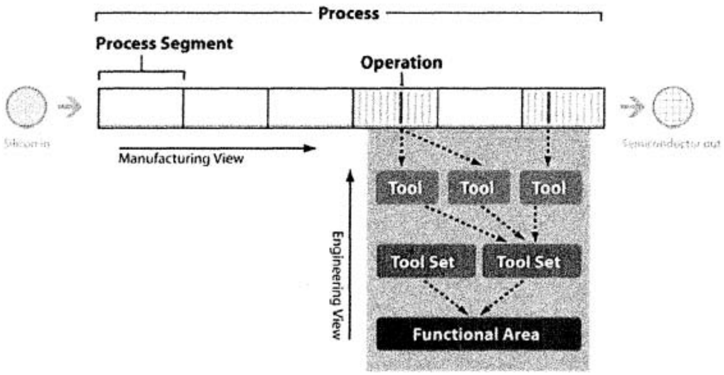


Fig. 2. Relationship between Process-Flow and Functional Area

Process flow is a manufacturing concept while functional areas are related to engineering concerns. This relationship indicates how the conflicting goals are disseminated throughout the system.

3.2 Social Organisation

Responsibility for controlling the Fab is spread across the social organisation of workers. Two of the main structures within the social organisation are manufacturing and engineering, mirroring the conflicting goals mentioned above. Within these structures a management hierarchy exists. Factory floor workers focus on smaller parts of the overall system and have limited information requirements for carrying out their jobs. Management level workers make decisions that relate to larger portions of the overall system. These decisions are directly affected by the production/quality conflict.

3.3 Development of Whole System Interface

Currently the system is operated through a range of individual applications. These have been developed to support specialist operations and provide only partial views of the overall system. Control applications exist at the tool level providing data relating to tool performance. While this allows an operator to control their tools, it is difficult to relate the data back to higher level metrics that inform us about the overall system state. The result of this set-up is that information must be combined from multiple sources to gain an understanding of system performance at higher levels of abstraction. Currently this is done through the manual generation of reports and verbal communication at management meetings.

The current information systems support the temporal control of system components. We can think if this as enterprise level SBB. However, enterprise level KBB cannot be supported through these systems as there is no interface that explicitly represents the relational structures of the fab. Such an interface would have a number of benefits over the current reporting tools. It would make it easier to see the effect of the conflicting goals at higher levels of abstraction, it would enable users to quickly drill-down and locate the cause of these conflicts and it could allow users to recognise patterns in system behavior. Here we attempt to use the EID framework to generate such an interface.

4 Development of ADS for HVM

We have discussed how the ADS for the DuressII system was generated by combining a functional abstraction hierarchy with a physical decomposition. In the Fab environment the physical decomposition has limited use. While physical tools match functional operations at the lowest levels, circulation in the process-flow means that physical and functional relationships no longer equate at higher levels of abstraction. In figure 2 we can observe that values associated with a process segment have no relation to values associated with a toolset. If a physical decomposition is no longer valid what other constraints can we use? The decomposition hierarchy is not limited to physical constraints. Other examples of system constraints that may be used include functional purpose, organisational and legal constraints [8].

4.1 Two Decomposition Hierarchies

We have already outlined two functional constraints of the system that are common across HVM environments. Both of these have had a direct effect on the organisational structure and both can provide decomposition hierarchies. The manufacturing hierarchy organises the system into different levels of granularity according to product position in the process-flow. The hierarchy consists of a process which is divided into segments which in turn are subdivided into operations. This facilitates a horizontal view across the process-flow. The engineering hierarchy allows us to think about the system in terms of equipment. It provides us with a vertical view down into functional areas, toolsets and tools (fig. 2).

4.2 The Abstraction Lattice

Both of the views mentioned above are valid system decompositions and can be used to generate independent ADSs of the system. However, as conflicting goals they are non-analogous. Items in similar positions in their individual ADS structures may not be related. How can we integrate these into a single model of the system that displays both structures? While they are very different at the abstraction level of functional purpose, they share the same properties at the level of physical form. This commonality can act as a bridging point between the two views. This allows us to develop our model as an Abstraction Lattice (fig.3). An Abstraction Lattice allows us to reason our way down through levels of abstraction in one view and then up through levels of abstraction in an alternative view of the same system. This approach allows us to reflect the Abstraction Hierarchy across the level of physical form joining up the two ADS representations. Our new ADS (fig.4) captures all of the system variables from both view at multiple levels of abstraction. This completes the first stage of the EID process. This ADS gives us a field description of the system and can be used to carry out use-case scenario mappings for various tasks [14].

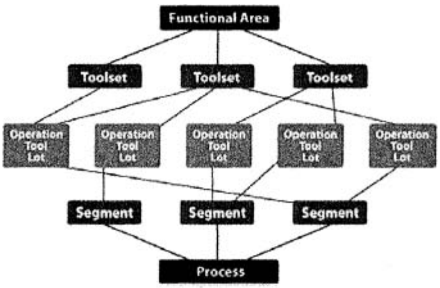
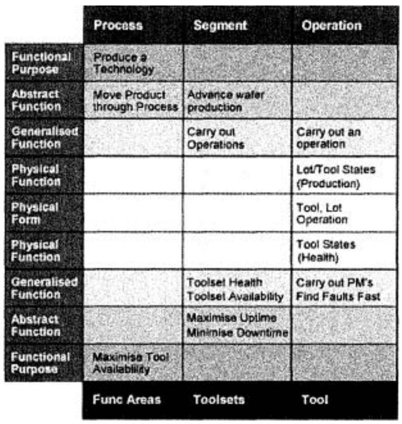


Fig. 3. The Abstraction Lattice



	Process	Segment	Operation
Functional Purpose	Produce a Technology		
Abstract Function	Move Product through Process	Advance wafer production	
Generalised Function		Carry out Operations	Carry out an operation
Physical Function			Lot/Tool States (Production)
Physical Form			Tool, Lot Operation
Physical Function			Tool States (Health)
Generalised Function		Toolset Health Toolset Availability	Carry out PM's Find Faults Fast
Abstract Function		Maximise Uptime Minimise Downtime	
Functional Purpose	Maximise Tool Availability		
	Func Areas	Toolsets	Tool

Fig. 4 . ADS for Fab

4.3 Information Requirements

Our goal is to facilitate KBB in relation to the overall system. To allow for this we need to supply an interface that can achieve a number of goals. Firstly, it must communicate information relating to both views of the system. Secondly, it must represent information from these views at different levels of abstraction. Thirdly, it must allow the user to combine information across the views at different levels of abstraction. In the next section we show how this modified ADS requires a different approach to visual representation than the approach used with the DuressII system.

5 Design of Visual Form

As previously discussed, the DuressII system used data chunking to embed the abstraction hierarchy into a visual design. A number of factors made this possible. Firstly, the ADS of the system used a physical decomposition hierarchy. In the interface subsystems were indicated through the physical clustering of components and traversal of the ADS was made possible through visually focusing on specific areas of the overall display. Secondly, the relatively small scale of the system meant that it was physically possible to display all of the variables in a single screen.

5.1 Differences between Microworld & Real World Applications

The Fab is a much more complex system. As we have seen, circulation in the process-flow means that a physical decomposition does not provide us with a useful model of system functionality. Our ADS uses two decomposition hierarchies based on the conflicting functional constraints of the system. While these hierarchies are related at certain levels they are not analogous and cannot be combined into a single graphic form, therefore both must be embedded in the display if KBB is to be supported. The scale is also different. While microworlds deal with a small number of variables, the fab features thousands of variables of different data types. Limitations of perception and cognition make it difficult for humans to work with this number of variables of mixed data types and complex structures. Bertin's "impassible barrier" indicates how it is impractical to represent relationships in data with more than three variables in a single image [4]. Furthermore, if we are to allow for direct manipulation of these variables, there is a lower-limit to the physical size we can represent them. This limits the number of variables that can be represented on a single screen. In order to represent the data we need to hide information within the levels of abstraction supplied in the ADS. Higher-level metrics or summary data can act as gateways into specific lower-level data. A challenge that arises with this is how to avoid "keyholing" [1], where the presentation of detailed information hides its context in an overall system.

5.2 Properties of Final Display

The differences in scale and complexity mean that visual chunking cannot be used as a technique for embedding our ADS in our final interface. This is further supported by the fact that our views are non–analogous, meaning that both hierarchies must be represented. We have defined three goals that our final interface must achieve if it is to support KBB.

1. Both abstraction hierarchies must be displayed
2. The hierarchies must be differentiable and compatible
3. Cross-hierarchy relationships must be made explicit

6 Representation of Hierarchies

Hierarchies are organisational systems that allow us to think about data at higher levels of abstraction. Different types of hierarchies include taxonomies, organisational structures and file systems. Tree structures, consisting of nodes, connections and leaves are the most basic way to visualise hierarchies. Nodes are organisational structures that can contain other nodes or leaves. Connections indicate the relationship between nodes. Leaves are low-level data that cannot be subdivided. Two core concepts have been used to define the graphical representation of hierarchies', connection and enclosure. [15]

6.1 Visual Hierarchies

Connection (fig. 5) uses the most literal visual representation of the tree structure. Here nodes are linked to sub-cases by lines indicating the connections. The structure is encoded on the spatial axes with the x axis carrying the nominal variable (N) and the y axis indicating the ordinal variable (O) of hierarchical level. The use of the spatial axes means that the representation can get quite unwieldy when dealing with large hierarchies.

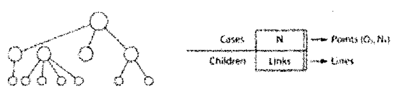


Fig. 5 . Connection Tree Structure [15]

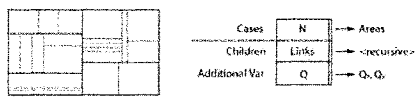


Fig. 6. Enclosure Tree Structure [15]

Enclosure uses area to represent nodes. Child nodes are contained within parent nodes in a recursive manner. Thus connections are indicated through enclosure. This approach allows us to use the area of nodes to convey quantitative data (Q) associated with the system. Again this techniques runs into difficulty when displaying large hierarchies as the deepest cases become very small and difficult to see.

The visual chunking of components in the DuressII system is a form of the enclosure technique. The need to represent quantitative component variables makes the connection technique unsuitable. Given the existence of two hierarchies we propose a dual display approach for our interface allowing us to divide up these representative tasks. This allows us to think about the visual representation of each hierarchy independently.

6.2 Suitable Representation is Defined by Task

The concepts above deal with visually representing the structure of a hierarchy. Having a view of the overall structure of a hierarchy is important for developing a full understanding of the system it represents, however this is only one use of it uses. The visual representation of hierarchies and their contents facilitates a number of tasks including: perceiving balance and connectivity of structures; making comparisons and navigating between levels; transferring content between levels; making comparisons and navigating within a level and; transferring content within a level. While many different techniques for displaying hierarchical information have been suggested (fig.7) they tend to support different tasks and actions to varying degrees of success. Here we review a set of common techniques and indicate the tasks they can support.

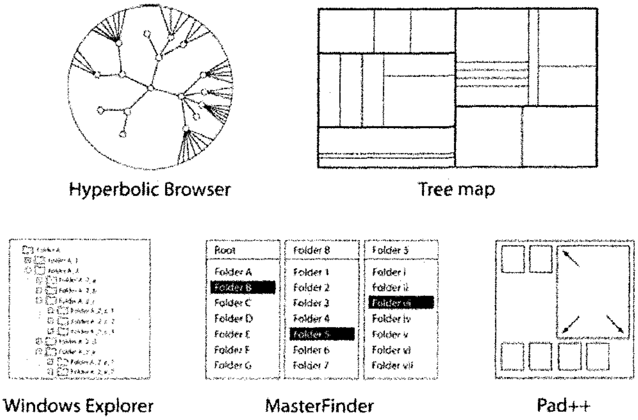


Fig. 7 . Hierarchy representation techniques

Hyperbolic Browser

Presentation technique: Graphical Tree
Nodes: Boxes, Connectors: Lines, Data: Not Represented
Tasks Supported: navigation, perceiving balance and connectivity
Restrictions: Shows overall structure but cannot carry additional metrics. Data must appear in a different window.

Tree map

Presentation technique: Nested boxes
Nodes: Boxes, Connectors: enclosed areas, Data: Area
Tasks Supported: Comparison within and across levels.
Restrictions: Recursive nature of display makes it difficult to see low level data.

Windows Explorer

Presentation technique: Semi-graphic Tree
Nodes: Folders, Connectors: Lines, Data: Not Represented
Tasks Supported: Browsing, Content Transfer
Restrictions: Only reveals a path in the overall structure. Use of labels makes it difficult to display the total structure. Data must appear in a different window.

MasterFinder

Presentation technique: Hierarchical Browser

Nodes: Window, Connectors: Adjacent window, Data: Filenames
Tasks Supported: Navigation, Content transfer between levels
Restrictions: Only shows path and related nodes through a hierarchy.

Pad++
Presentation technique: Zoomable User Interface
Nodes: Boxes, Connectors: enclosed areas, Data: text or graphic
Tasks Supported: navigation, Comparison within and across levels.
Restrictions: Zooming makes comparison of low level data difficult.

6.3 Analysing our Case Study Hierarchies

No single representation supports all possible tasks related to hierarchies with maximum efficiency. In order to understand what representational technique is best suited to our problem we analyse the hierarchies involved in our ADS. We examine each view under the following three terms: purpose and task, data involved and depth of hierarchy. Purpose and task can inform the general technique we use for representing the hierarchical structure. Knowledge of data types can be combined with rules of data representation [4, 5] to further inform the display. The depth of the hierarchy can indicate whether a list-style presentation of the hierarchy is possible.

6.3.1 Process view

Purpose & Tasks: This manufacturing based view is a horizontal view across the process flow. The purpose of the view is to see the volume of product in the plant. This should allow the user to see the productivity level of toolsets, segments and the overall fab. The view at the lowest level allows the user to spot potential bottle-necks in the process-flow. The tasks involve viewing product levels at an operation (toolset), comparing product levels between toolsets, viewing product distribution across the process, viewing product capacity levels in segments and viewing product capacity across the entire fab.

Data Involved: The structure of the hierarchy is based on the division and subdivision of the process flow into more manageable units namely process, segment and operation. As a part-whole hierarchy it is possible to think of operations as having a quantitative relationship to both its parent segment and the overall system. As the process flow is a series of consecutive steps the nodes within each level can be said to have an ordinal relationship with each other. The additional data variable to be displayed is volume of product. This is quantitative data.

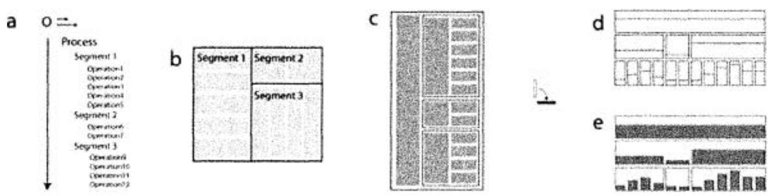


Fig. 8. Display design of process view

Depth of Hierarchy: The hierarchy is shallow consisting of only 3 levels process, segment and operation.

Design Rationale: With only three levels of depth the process hierarchy is relatively shallow and may be easily flattened. This allows us to convert it into a list representation making it easier to understand [16]. The ordinal relationship between nodes at all levels further supports this list representation as the use of a spatial axis supports ordinal data (fig. 8a). However, a list style presentation is not optimal for displaying the quantitative values associated with product volumes. Having segments interspersed with operations makes it difficult to carry out quantitative comparisons at these different levels of abstraction. An alternative approach is to use a treemap (fig. 8b). Here the higher level metrics are created naturally through the use of enclosure. Unfortunately the display then loses the ordinal relationship between the nodes necessary for comparison. A solution is a hybrid between the two. The enclosure technique is combined with a list to give a display that allows for both comparison at a level and between levels of detail (fig. 8c). The volume metric can then be expressed through position of point or area as both of these visual variables support quantitative information [4].

6.3.2 Functional View

Purpose & Tasks: The second display, the engineering based view, is a vertical view down into the system. The purpose of the view is to analyse the performance and availability of equipment across the fab. This involves looking at data from individual tools and toolsets and understanding the relationship between lower level data and higher level metrics. The tasks include accessing tool data, comparing tool data, controlling tool activity, transferring tools between toolsets, accessing and comparing toolset metrics.

Data Involved: The structure is a taxonomic hierarchy based on functional activity. This means that the relationship between nodes both vertically (down through the levels) and horizontally (across a level) is nominal. While different functional areas will have different numbers of toolsets, tools may operate at different speeds (throughput rates) so proportional or quantitative relationships between toolsets and functional areas are not very relevant. While the process view only needed to carry one additional variable (volume), functional view nodes have a number of associated variables relating to performance and availability at all levels. Some of these variables can be calculated into higher-level metrics relating to parent nodes. For example the health of a toolset is derived from the health of its associated tools which in turn derive their health from a set of sensor readings.

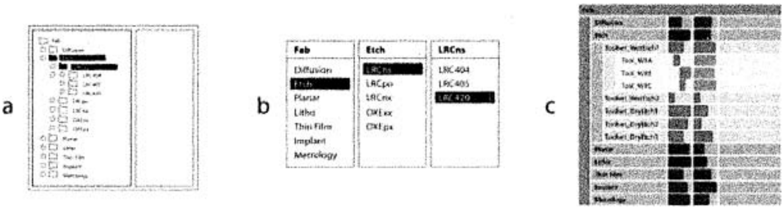


Fig. 9. Display design of functional view

Depth of Hierarchy: The functional hierarchy has two further levels not represented on our ADS. At the top level we have the overall Fab and at the lowest level we have individual tool variables that give us the tool health figures. This is a deeper structure consisting of five levels fab, functional area, toolset, tool and tool variables.

Design Rationale: As the structure is a deep taxonomical hierarchy it should not be flattened into a list form. The structural organisation is nominal not ordinal so a treemap type approach is possible. A treemap approach could also support a health metric through area and an availability-metric through colour. However, one of the tasks is the comparison of attributes at different levels. The arbitrary shapes generated by the treemaps space-filling recursive algorithm, coupled by the small display sizes at the lowest levels of granularity makes this comparison difficult. What we require is a path through the hierarchy displaying metrics at each level of granularity. The windows explorer (fig. 9a) allows us to unfold hierarchies through interaction, but it only displays data for one active node at a time and this appears in an adjacent window. The hierarchical browser (fig. 9b) also unfolds a hierarchy but it achieves it by opening a new adjacent list of nodes each time a node is accessed. Our display needs to combine the information hiding techniques of the windows explorer with the display capabilities of the hierarchical browser. However rather than displaying all of the associate nodes in the path we display only the metrics for the nodes associated with the path. By aligning these metrics along a horizontal axis it should become possible to read and compare performance across levels in the hierarchy as well as drilling down into the hierarchy to reveal the explanation for the metrics (fig. 9c).

6.4 Making Views Compatible

Achieving Knowledge Based Behavior requires information from both views of the system and at various levels of abstraction. We identify three qualities that our representations must have in order to achieve this. The representations must allow the user to:

1. Easily differentiate between the two hierarchies
2. Relate low-level data to higher level metrics
3. Identify relationships between views

6.4.1 Visually Differentiate between Hierarchies

The abstraction hierarchies for both views are similar in form and scale. This makes it possible to carry out use-case mappings and reveal information requirements [14]. However, the similarities must end here. The two views relate to different information and are used in different ways. Workers in manufacturing have a greater interest in process flow while engineering is more concerned with the system health. While many workers will benefit from having both views, it is critical that their visual representations make them easily discernible. There are a number of reasons for this.

Firstly, similar representations would require labeling and the reading of labels to allow for differentiation. This increases the cognitive work load of the user and would reduce performance. Secondly, a similar rendering of the hierarchies would give a false impression of correlations between levels of abstraction. This would give a false interpretation of the data relationships as the hierarchies are non-analogous and this correlation does not exist. Thirdly, a resemblance between the views could lead the viewer to access the wrong information. An example of this would be browsing through a process view when looking for health information.

We have seen that there are many different ways to visually represent hierarchies. It is important that we select methods that are visually different to avoid the problems listed above. In our case the tasks associated with the hierarchies and the data involved have already guided us toward different visual representations.

6.4.2 Relate Low-level Data to High-level Metrics

Abstraction hierarchies allow us to structure complex systems and help in the development of conceptual models. They also enable us to abstract low-level data into higher level metrics allowing us to view system state information at different levels of detail. It is important that the visual representation allows us to understand the hierarchies involved. Different display techniques can achieve this in different ways.

In the proposed process view (fig. 8) we need to represent the product volumes at the level of operation, segment and overall process. Volume is a quantitative variable and can be represented through either spatial position (fig. 8d) or area (fig 8e). While the spatial position of a point is a valid representational format, its small surface area makes it difficult to see volumes across multiple areas. The other alternative, using area, allows us to see waves of product volumes across the process. It allows us to make visual estimations of volumes between segments and can also be used at the level of segment and process to indicate product levels with regard to a maximum volume.

The functional view is somewhat different. Multiple variables exist at each level of abstraction. These relate to a number of different metrics including availability (time to next scheduled maintenance), health and health history (fig. 10). All of this information is important for diagnostic tasks allowing a user to search and compare data in order to find causes of faults. Readings at any specific level may be horizontal or vertical. For example a user with responsibility for the etch area may look across the etch data and see that the health level is lower than expected. By clicking on the etch label the user can expand the matrix to reveal the lower level toolset data. Now, by scanning vertically down the health readings the user can identify the toolset with the lowest health. By repeating these actions on the toolset the user can identify the tool(s) responsible for the low health reading and act accordingly. This representation combines the advantages of the hierarchical browser with a matrix arrangement. Information relationships across levels of abstraction can be made explicit by using a common visual variable and spatial arrangement. The result is a display that allows us to hide detailed data behind higher level metrics but can also reveal information at multiple levels of abstraction by folding out the matrix.

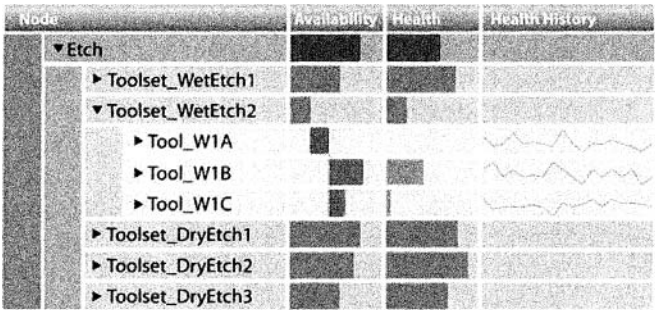


Fig. 10. Relating data to higher-level metrics

6.4.3 Identify Relationships between Views

The views offer two alternative ways of looking at the system. While they are non-orthogonal, certain relationships exist between them that are important for decision making. For example, the volume at an operation contributes or correlates to the volume at a toolset, so knowing the health of that toolset is helpful if manufacturing decisions are about to be made. Similarly, if a tool is about to be shut down for repair, it is important to know the health of the other tools in the toolset and the volume of work that is about to arrive.

We make these relationships explicit through the use of contextual highlighting. On selecting a functional area, the related operations in the process flow are highlighted (fig. 11). This allows the user to see how product levels are related to the health of that area. As a user drills down through the functional hierarchy, the number of related operations decreases making it easier to see this relationship. The flattened presentation of the process hierarchy is helpful as functional activities are scattered throughout the process. From the other perspective if the user notices a peak or a trough in the process flow, rolling over an operation will highlight the related functional area.

Fig. 11. Making relationships explicit

6.5 Make Relationships Explicit

To support Knowledge Based Behavior the user must be able to fully navigate both hierarchies. This allows them to gain a full understanding of the system state. This movement may occur in different ways depending on the task being carried out. It may be synchronous between levels of abstraction in both views, for instance comparing toolset health and product volumes at an operation. At other times information may need to be combined from different levels of abstraction as with the health diagnosis scenario discussed above (section 6.4.2).

A representation that displays multiple levels of abstraction allows us to visually jump between the metrics. This is the case with the process view where we can see product volumes at both operation and segment levels. Alternatively, interactive techniques can show contextual information on demand. The unfolding matrix in the functional view hides lower level data behind higher level metrics until they are requested. While these techniques allow us to navigate either hierarchy independently, the use of contextual linking could allow us to navigate them together making their relationship even more explicit.

6.5.1 Contextual Linking

We have already pointed out how contextual highlighting allows the user to identify relationships between views. Contextual linking would further extend this by allowing interaction with one hierarchy to control the display of the other. Above we describe how rolling over an operation in the process flow can highlight the related functional area. A contextual link would drill down into the functional view to the appropriate level of detail. For instance, when the user clicks on an operation in the process flow the engineering view would expand to display the related toolset.

This has further implications for the design of the process view. If the process flow operations are to be clickable there is a lower limit to the size that can be used to represent them. In the current design, area is used to encode product levels. However, at very low levels of production this area may be too small to act as a target. If we examine the activity being carried out on the display we can see that there is another option. Users are viewing the product volumes to note peaks or troughs in the data. Their task is not to estimate quantitative differences between adjacent operations but rather to understand the ordinal relationship between groups of operations. This is a simpler task and requires less detailed information. We carry out a scale transformation on the data turning detailed quantitative data into a series of ordinal ranges. Ordinal data can be supported by a wider select of visual variables including tonal value or shade [4]. By fixing the area of the operation and encoding the product volume using tonal value we can provide the user with a clickable representation of the process view (fig. 12) thus allowing for contextual linking.

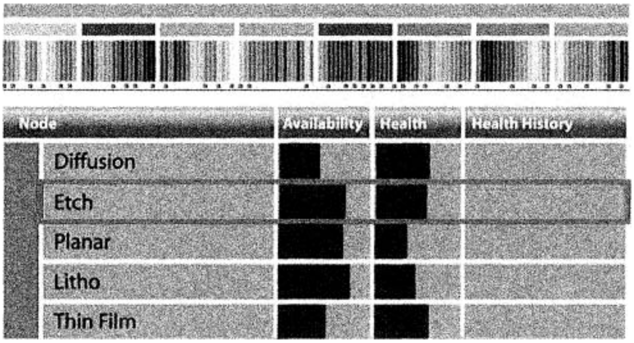


Fig. 12 View from final visualisation

7 Conclusions

EID provides us with a useful framework for designing interfaces for complex socio-technical systems. However, its application to large scale systems is not as straightforward as the approach given in the original framework.

In its first stage, the use of physical decomposition in the ADS is only useful where the physical constraints of the system match its process-flow. The introduction of circulation into process flows makes this approach unsuitable and requires functional constraints to be used instead. The manufacturing and engineering functional constraints associated with HVM are conflicting and cannot be integrated. Instead a modification of the ADS can allow us to integrate two abstraction hierarchies into an abstraction lattice.

In the second stage, we note that the "principles of visual display" discussed in section 2.2.2 are easily applied to smaller domains but less easily achieved with large complex domains. The scale of the Fab and the lack of a physical decomposition makes visual chunking unsuitable for embedding the ADS. Instead we must use more advanced visual techniques to represent our Abstraction Lattice.

Three major goals must be achieved to support KBB in the final interface. Firstly, both abstraction hierarchies must be explicitly expressed. Secondly, the hierarchies must be differentiable and compatible. Thirdly, cross-hierarchy relationships must be made explicit.

A design rationale has been produced which draws on the ADS to achieve these goals. Each hierarchy can be independently analysed under the terms of tasks, data types and structural depth. This information can be combined with guidelines for graphical data representation to inform the manner of hierarchical representation. Relationships between the two hierarchies can be established using a combination of visual organisation and interactive techniques.

8 Acknowledgements

This work has been funded through an Intel Ireland Research Scholarship. The Authors would like to thank Michael McGrath and Joe Butler at Intel® IT Innovation Centre, Intel Ireland, Leixlip, Co. Kildare, for their valuable assistance.

9 References

1 Woods, D., Johannesen, L, Cook, R., Sarter, N. (1994) Behind Human Error: Cognitive Systems, Computers and Hindsight CSERIAC
2 Zhang, J. & Norman, D. A. (1994). Representations in distributed cognitive tasks. Cognitive Science, 18, 87-122.
3 Hutchins E. (1995). How A Cockpit Remembers its Speeds. Cognitive Science, 19, 265-288
4 Bertin,J. (1983) Semiology of Graphics The University of Wisconsin Press, Madison. Translated by William J. Berg.
5 Zhang, J. (1996). A representational analysis of relational information displays. International Journal of Human-Computer Studies, 45, 59-74.
6 Zhang, J. (1997). The nature of external representations in problem solving. Cognitive Science, 21(2), 179-217.
7 Potter, S. S., Elm, W. C., Roth, E. M., Gualtieri, and J., Easter, J., (2002). Bridging the Gap between Cognitive Analysis and Effective Decision Aiding. In M. D. McNeese and M. A. Vidulich (Eds) State of the Art Report (SOAR): Cognitive Systems Engineering in Miltary Aviation Environments: Avoiding Cogminutia Fragmentosa! Wright-Patterson AFB, OH: Human Systems Information Analysis Center. (pp 137- 168). Also available at: http://iac.dtic.mil/hsiac/.
8 Vicente, K. J., and Rasmussen, J., "Ecological interface design: Theoretical foundations," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-22, pp. 589-606, 1992.
9 Bisantz, A. M., and Vicente, K. J., "Making the abstraction hierarchy concrete," International Journal of Human-Computer Studies, vol. 40, pp. 83-117, 1994.
10 Vicente, K. J. (1999). Cognitive Work Analysis: toward safe, productive, and healthy computer-based work. Mahwah, NJ: Lawrence Erlbaum Associates
11 Rasmussen, J. (1979). On the structure of knowledge–A morphology of mental models in a man-machine system context. Risø Report M-2192. Roskilde, Denmark: Risø National Laboratory, Denmark
12 Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. IEEE Transactions on Systems, Man, and Cybernetics, 15(2), pp. 234-243.
13 Rasmussen, J (1983). "Skills, rules, knowledge: signals, signs, and symbols and other distinctions in human performance models." IEEE Trans. Syst., Man, Cybern. Vol. SMC-13.pp.257-267.1983.
14 Upton, C. and Doherty, G. (2005) Designing Usable Decision Support Systems for HVM. Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation, Vol. 1, pp. 459-466 Lo Bello, L. and Sauter, T.(Eds.)
15 Card, S., Mackinlay, J. and Shneiderman, B. (1999) Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann
16 Waloszek, G. (2004) "Fighting (with) Hierarchies – Part II: Presentation" SAP AG, Product Design Center – 04/27/2001, updated 01/29/2004. available at : http://www.sapdesignguild.org/community/design/hierarchies2.asp

VIHO
- Efficient IT Support in Home Care Services

Niklas Johansson and Bengt Sandblad
Department of Information Technology,
division of Human-Computer Interaction,
Uppsala University, Sweden
{niklas.johansson, bengt.sandblad}@it.uu.se

Abstract. The main research objective for the VIHO project (Efficient Computer Support in Care for the Elderly) was to investigate how a home care and help service organization can be developed in order to be better prepared for future challenges, and how new technical systems could support the development process. We have studied the home help personnel's need for support and investigated how the new organization could be formed in order to provide a healthy and sustainable work. Initial focus has been on the essential parts of the work and how the work could be improved in the future, and not on design of the technical support systems. Our basic point of view has been that correctly used, new technology can contribute, so that work and organisation develops in a positive way, patients are feeling secure and the personnel's work environment is improved. This means that the organisation better can fulfil expectations and requirements. The professions can be strengthened and the organisation will be able to meet future challenges. In this report we briefly describe the results and the methods used in the project.

1 Introduction and Background

When an IT- system is introduced in a work situation, this will result in a number of consequences. Both the organisation and the work practices will experience substantial changes. The work tasks will not be performed in the very same way as before the introduction, since IT-systems by its nature facilitates, supports or sometimes prevents certain ways of working.

To obtain positive effects of introducing new IT-systems at a work place, we find that it is necessary to also develop the organisation and the work procedures. The work that is to be supported must be developed and improved prior to introduction of new technology. We find it not enough to use modern IT-systems to support work in

Please use the following format when citing this chapter:

Johannson, N., Sandblad, B., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 47–66.

the way it is currently performed, instead it is more important to see the potential in how the new technology can contribute to a positive development of the work and the organisation as a whole. In other words, we do not want to "cement today's cow paths", i.e. focus to hard on how the work is performed today. Our goal is to use technology in order to obtain positive changes, making the work more efficient and the work environment better.

The organisations for care of the elderly are today struggling with a number of problems and challenges. Work load is increasing since more people are becoming even older and are in an increasing need of attention and care. Different care professionals with different responsibilities will work together in a new ways. The economical resources within the welfare service are limited and impose requirements of effectiveness. Care takers themselves, and their relatives, are demanding more participation, information and service of a high quality. Nevertheless, there must be qualified personnel, sincerely devoted to the work, that can and want to work within the care sector in the future. This results in requirements of a good and healthy work with proper possibilities for development, efficient tools and a good working environment.

This report describes how we, base on previous research, have developed present work systems within care for the elderly and home health care, and proposed a new, future, work practice supported by new types of IT-systems. The project is called VIHO, a Swedish acronym for "efficient IT support in care for the elderly". Within this project we have used, and further developed, the future seminar model (Vision seminar) presented by [1]. Result from this work is manifold; the Vision seminar method has been improved, new work organisation and processes have been proposed to the home health care services and a preliminary design of a mobile IT system has been suggested.

2 Research Framework

Our approach is based on a number of basic values and fundamental conceptions that throughout have influenced our work. The essence of our relation to design of supporting IT-systems can be found early in the development process. First we design the work practice, then the support systems. Main concepts in our scientific approach are largely inspired by action research and a user centred method of working with description, analysis and design of IT-systems.

2.1 Work Environment Aspects and IT Supported Work

The way IT-systems are designed, and how they are used in practice, has a large influence on the working environment and the users' health. Issues of work environment have several dimensions and can be divided in three different parts:

Physical work environment concerns the ergonomy of the work environment. The work place must be designed to facilitate a healthy work. IT-supported work imposes requirements on the equipment. The monitor, keyboard, mouse, chair, table, lighting etc. shall all be ergonomically designed and adopted to the specific user and his or hers work situation. A poor physical work environment leads among other things to eye problems, strain injuries e.g. pain in neck, shoulders, arms and hands. There exists much knowledge about such problems [2].

Psychosocial work environment is the corresponding mental apprehension. It concerns how the personnel feel at work and how they experiences their work situation, including internal relations, relations with the management, provided and experienced social support etc. Bad such conditions do mostly result in a feeling of lack of support, stress and feeling ill at ease. IT support can affect such factors in a negative way if they are not carefully introduced. Examples of negative effects from IT are bad designed functionality for monitoring, time measurement and distribution of workload.

Cognitive work environment concerns the match between the IT-system and our and our cognitive skills. An example is when something in the work situation prevents us from using our knowledge and skills in an effective way in order to perform a task. A computer application can for example hinder from effective work, making the work problematic, difficult and hard to understand. It takes long time to perform, errors are frequent and the technology is confusing. It is important to understand these problems as they can affect the work negatively by making the personnel ineffective, insecure with feelings of irritation and stress as a consequence. Other important problems that can arise when using supporting IT systems in work are:

- Restrains and lack of freedom. The user becomes deskbound at the computer large parts of the day, mostly in static, monotonous positions.
- Controlled, a feeling of being governed in the work by the IT system, i.e. a lack of possibilities to control the system.
- Stress and the feeling of high demands from the work in terms of work load, time pressure and badly functional IT support are exceeding the resources at hand and the personal abilities.
- Stress related psychosomatic conditions, i.e. headache, annoyances, stomach-ache, lack of sleep.
- Physical problems, foremost ache in the back of the head, shoulders, arms and hands for example "mouse elbow".

It is important to recognise the strong interaction and relationship between IT systems, efficiency in the work process and the work environment. The supporting IT system must support efficiency in the work process and a healthy work environment. For a user, the largest work environment problem can be the feeling that you do not work efficient enough. Users are often much more concerned about the efficiency that e.g. about the physical work environment. Usability is in other words not only a question about work environment, but concerns very basic requirements for an efficient and sustainable work.

2.1.1 Healthy Work

An illustrative model describing healthy work was developed by Karasek and Theorell [3]. It describes the work situation in relation to the subjective experiences of demand, control and social support at work. Demands are the requirements laid upon the worker, what he or she is obliged to do. Control is the means the worker has to survey and handle the work situation. Social support refers to overall levels of helpful social interaction available on the job from co-workers, supervisors and other recourses. The primary work related risk factor is the workers' lack of control over how to deal with the job's demands and how to use their skills [3].

High demands are in other words not a problem, if they are combined with a high control and strong social support. But if the demands become too high in relation to control and social support, this will lead to high stress and an increased risk for bad health. The most favorable situation is one characterised by reasonable to high demands, high decision latitude and a strong social support [2].

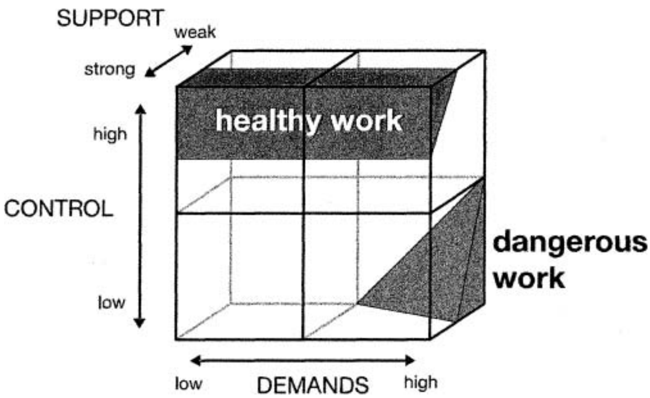


Fig. 1. Karasek and Theorell's model describing healthy work

Our experiences show that an increased computerization at a work place often results in experienced higher requirements [2]. This is by itself not a critical phenomenon, but it must be counterbalanced with increased control and social support. Today our experiences are the contrary; experienced control together with social support is often reduced together with an increased responsibility. This will lead to major problem in the work environment as well as increased risks for bad health and illnesses. When new work processes are designed, or when new technology is introduced, we must strive for increased control and social support.

2.1.2 Insufficient IT Systems

Today we can se several examples where new IT systems are introduced in work places, and how these systems unfortunately do not work properly in practice [4]. If it is possible to perform a work task in a more simple way than the one proposed by the IT system, the user will choose this simpler solution. Some reasons for an IT system to be neglected and not used as intended are:

- Bad adaptation of the system to local conditions at the work place

- The system only provides support to some parts of the work processes
- Integration of a specific IT system with other involved systems is poor
- The system is difficult to use and it is characterised by bad usability
- The system does not give a proper feedback, not providing the user with an understanding of the system's benefits.
- The system does not support, but rather prevent, development of the work organisation
- Poorly technical performance makes the system slow and difficult to handle.

It is our belief that the required increase in control and support can be facilitated by a well designed and supporting IT system. Routines for support and control must also be well established in the organisation. Well designed and introduced IT systems can provide the user with efficient support like for example:

- Effective access to updated information
- Safe and secure storage and access of information
- Overview of the work situation and complex information sets
- Useful and efficient communication
- Control over the work situation through usable interfaces
- Control over the work situation through efficient planning tools
- Possibilities to evaluate the work process through access to historic information and evaluation tools
- Individual and collective development of competencies and work organisation

2.2 Action Research Approach

Our research is practical, performed in working life and can be considered as action research [5]. In the project described in this report, we have initiated a process in order to answer our research questions about how work processes and IT-systems can be jointly developed in practice. The objective is not to develop a new IT-support system for the existing work organisation and work processes. The objective is rather to use the potential of new mobile technology to enhance the development of the work and its organisation as such. In this way we can fulfil two different goals. First we can make sure that the new IT-support system is well adapted to the new work processes. Secondly, we can use the potential of the new technology to support a good development of the work, and not to preserve old structures and processes.

A limitation with the action research approach is that the method can not be strictly evaluated. The conclusions can only be made whether the tested processes were (eventually) successful during the present conditions or not.

In comparison with an ethnomethodological approach, action research also aims at studying the subject in a connection with the organisation and its personnel. Action research aims at making a change, while ethnomethodology focuses on understanding. However, in order to propose a successful change, we find it important to first achieve understanding.

In our research we strive for an improved way to develop and introduce new supporting IT-systems at work places, and how to gain valuable knowledge about how this can be done in practice. To achieve this, we find that action research is a suitable approach.

2.2.1 New Work Procedures

As discussed briefly in the introduction chapter, an introduction of a new IT system in an organisation will undoubtedly change and affect the organisation and the way work is performed. The work will not be the same as before, but a new work has been formed. This change must be carefully handled.

A work is never perfect and performed in the very best and effective way. There is always room for improvements. When a supporting IT system is developed, that means that the IT system will be supporting a work that most likely has the potential to be changed to something better. Ineffective procedures can easily be made permanent. We find it better to develop IT systems that supports an improved way of working. In this way we can use technology as a driving force for work and organisational improvements.

We want to take control over the change of work that occurs when introducing a new IT system, and turn that change into something productive. Our approach is to first create a specification of a good way of working and thereafter design an IT system supporting that work. In this way we achieve two things; a new improved way of working and an efficient supporting IT system.

2.2.2 Participative Design of IT Systems Supporting Work Processes

Another main foundation in our work is that the ones that truly know best how to improve the way of working and what kind of support an efficient IT system can provide, is the personnel from the organisation, i.e. the ones that are to use the IT system as a tool in their future work. In line with a user centred system development approach [6], development of good, healthy and efficient IT systems are to be performed in close cooperation with the future users of the system.

A related research approach is participatory design [7]. There is a lot of research done concerning participatory design and user centred methods in general, and we will here just briefly conclude the most important.

User centred development is based on the principle that all development of technical IT systems for a workplace must be performed through active participation by persons that know the organisation and the work practice, i.e. the ones that actually performs the work. Experts from the outside can never fully understand the activities and can never alone decide whether a solution is good or bad. The ones that truly know the workplace and its practice can on the other hand not by them self describe and analyse their own organisation and work practice. Neither do they have the full competence to propose new, innovative solutions regarding the organisations as well as its IT support. It is only together and based on a suitable model for cooperation between future users and designers (researchers) that the work can succeed.

2.2.3 To Use the Users' Knowledge

Representatives of the organisation, the presumptive users, are experts in the way the work is performed today. They are the only ones that truly know the organisation in detail, details that are of grate importance when learning how systems and interfaces are to be designed. The way a work is actually performed in practice is not possible to completely describe, but has to be studied in practice. The work practice and the tacit knowledge that experienced professionals possess is very valuable and must influence the design [8].

Representatives also often possess valuable ideas of improvements, based on their experiences. However, it is our belief that these proposals should not be directly implemented, but have to be evaluated, expanded and arranged to fit in the new proposed way to work. New technical systems with its potential of information and communication facilitate totally new solutions. These possibilities must be worked out during the project.

As a part of the fundamental design of work, users' ideas and solutions to problems must be considered. But the designers are the experts of the final design solutions. On the other hand, no design solutions should be accepted without evaluation together with the users.

2.3 The Iterative, User Centred Approach

User centred system design does not imply that users themselves create the design of the IT system and its interface, but it aims at creating a creative environment and structure where designers and users can meet. In the VIHO project, our approach has been to create a basis to build further design work on. The basic main ideas behind the user centred model for system development are:

- Development work is conducted in collaboration between different competences, primarily between system developers and organisation representatives (users). They each have their own competences and area of responsibility but are parts of the same development work. The users are not service people or study objects, but is considered the part in the development project that possess the knowledge of how the concerned work actually is performed in practice. Participating users must be given required knowledge and the ones responsible for the design work must have good knowledge in human computer interaction and interface design.
- Development work is conducted in an iterative way where the phases analyse, design, construction and evaluation are repeated until the system has developed into a final version and the result is accepted by the involved parts. Representatives from the organisation must be different persons than the ones that are a part of the development work in order to minimize the risk of a misleading evaluation.
- Work is focused on prototypes of the planned system. A prototype changes during the project, from rough sketches on paper to an executable system. In this way it is important to use effective techniques for prototyping and evaluation.
- After implementation and deployment, the system must regularly be followed up and when necessary modified and complemented.

3 The VIHO project

Our overall goal in the VIHO-project has been to develop future supporting IT system. To do this as good as possible, we have first defined and developed the work itself. In this way we will jointly develop both the work and the IT system.

The town-district "Kortedala" is located in the north-east part of Gothenburg in Sweden. Together with personnel from Kortedala home health care sector, we have developed a vision for the future work within the health care area and how this work can be supported by new IT systems. This vision can provide a basis for development of the new work organisation in detail, to describe new work processes and to implement new efficient and usable IT systems.

3.1 Vision Seminars with a Work Group

Practical work of envisioning the future work at the workplace was made in what we call Vision seminars [1] together with a work group from Kortedala home health care organisation. Vision seminars are performed in order to envision future possibilities of changes of a work organization as well as practices and are performed as group discussions about present practices, problems, goals, and possibilities. The inherent goal is a more efficient, stimulating and sustainable work. Objectives of the process in the VIHO project were to provide a foundation for a sustainable organisation of work, redesign of jobs, and to propose functionality for a future supporting IT system that enhances the workers' skills. To make "the future" concrete and a bit more tangible, we have worked with a perspective of five years, which in the beginning of the seminar series makes year 2008.

The vision seminar series in the VIHO project was conducted during half a year with a working group of six assistant nurses, all experienced professionals from the elder care sector. Under the direction of four researchers, the group carried through twelve seminar occasions with two to three weeks between each occasion. During the seminars, the researchers role was to lead the discussions, provide input and to document and reflected upon the accomplish work. The reflections from one seminar served as an input for the next seminar.

At first, today's work was described in detail, closely followed by analysis of the needs for changes and development. Subsequently, important aspects of how the desirable future work would be carried out were formulated. These aspects included requirements of a changed work organisation, improved work processes and requirements of effective, supporting IT systems. Main objectives are to create a healthy and efficient work, at the same time as security and quality in health care are improved.

Fig. 2. Seminar with the work group

A fundamental approach has been to allow the process to take time. Meetings in the vision seminar series were held within two or three weeks of each other. This gave the participants time to reflect upon the previous meeting - consciously as well as unconsciously - and to prepare for the next meeting.

To facilitate the discussions and to prepare the participants for the next meeting, an assignment was often given for the next meeting. The participants were encouraged to engage their colleagues from their respectively workplace when completing the assignments. The distributed assignments were often related to how things are carried out, how and why, and they were closely coupled to the theme of the next seminar occasion. Here follows a number of brief examples of assignments:

- Write the essay "A day at work". Describe an average day filled with the activities that use to occur. Reflect upon what is good/bad, easy/hard, simpel/complicated, fun/boring etc. in the work.
- If your work place was your own company, and you were the manager, what would you do? How would you like your company to be organised and managed?
- What kind of communication are you carrying out daily? Why? Would you like some communication to be more extensive or more narrow than today? Why?

Every meeting had a theme, e.g. "Organization", "Technology" and "Information and communication". The working group discussed their work from different perspectives, the organisation, communication, skills, and collaboration and so on. Inspiration in the form of new knowledge on interesting topics e.g. organisation, ethics, and new technology were provided by internal or external lecturers. This put the participants in a better position to perform critical evaluations, creative thinking and to identify and propose solutions to problems at their own workplace. A number of seminar themes are here briefly presented.

3.1.1.1 Present Work

The aim of this analysis was to document present work, tasks, cooperation, and workers' skills, and it extended through a number of working group meetings. Present conditions were investigated initially. Skilled people easily describe the work they are involved in at present, what works well, and what does not. Rather than making a list of a number of tasks that workers perform, a more detailed description such as a told narrative can capture the essentials of work in its context. In order to get an overview of the present work situation, the seminar group discussed questions as for example:

- What do you consider most important in your work?
- What do you appreciate most in your work?
- What work do you carry out really well?
- What does good quality mean in your work?
- Which technical systems do you use? For what purpose?
- Which information do you use in your work? For what purpose? Where? When? How?
- With which people do you communicate? What matters do you discuss? How? Why?

When collaboration is considered, the complexity of the work routines at a workplace becomes evident. Such aspects easily remain concealed if work is regarded as a set of individual tasks that have to be managed.

3.1.1.2 Change of Work Procedures

To stimulate proposal to a better, desirable work, the participants in the seminar series were to play with the thought "If my workplace was my own company and I were the boss, what would I do"? The assignment was approached in a number of ways, for example concerning:

- Economy
- Competence, education and training
- Use of working time
- Scheduling
- Work planning
- Work evaluation
- Communication
- Information
- Leadership
- Rewards

In this way we achieved a good understanding of how the work was presently carried out in the different work places. Furthermore, a number of issues to change and improve were recognised and documented. It is our experience that discussions like these mostly provides good ideas. When asking a question like "What does really hinder this?", it used to turn out that there are very few restrictions for changes in practice, and that much is not very difficult to accomplish. Some things are even possible to change instantaneously.

A question raised during this seminar was about awareness of costs for medical supplies. After discussion of the issue, an assistant nurse from the home health care decided to label all the medical supplies in the storeroom at her work place with the cost for each item. This appeared to make the personnel aware of the cost and less predisposed to waste the expensive material unnecessarily.

3.1.1.3 Information and Communication Analyses

To carefully examine what kind of information is handled during the work procedures, we asked the working group participants to collect forms, lists and other documentations means from their work. Use of information were analysed by means of what information is used and what is shared. Communication then is much relative and was more extensively documented in terms of who communicates with whom, why, where, when and how.

In this work the gathered materials from the participants were very useful. When working with the material, it became clear that the participants in the seminar group quite often used different name for the same kind of form. They also used the same form in different ways - and this in the very same local organisation.

3.1.1.4 Scenarios

At the end of the vision seminar the group described the future work in form of a set of scenarios. A scenario is a written description, or story, describing - in this case - how the future work can be carried out. We wrote the scenarios iteratively together with the participants from the work group. The writing itself is valuable as it is a creative process where unformed thoughts have to be formulated in text.

Important parts of the scenarios were specified and described in terms of activity specifications. This is a more formally description of tasks that are performed, which information is used and produced, how persons communicate etc. Based on these scenarios and the activity specifications, we summarised the information and communication needs of the future work procedures.

Based on the written scenarios, simple and preliminary sketches of new supporting IT systems were carried out. In this way, the roughly designed IT-support systems are based on rather detailed descriptions of the proposed future work settings.

3.1.2 Complimentary Work

3.1.2.1 Interviews

The vision seminar process puts a grate focus on the participants in the seminar group. In order to not totally exclude other concerned professions, we conducted a number of interviews with key persons in the organisation. This was done in the latter part of the seminar series, based on the proposed new work taking shape. Results from the interviews where if necessary discussed during the seminars and included in the documentation.

These "external interviews" had two major purposes. At first we wanted to map other persons experiences. By exclusive concentrating on the people in the seminar group, other concerned professions were not allowed to be heard and we would risk getting a distorted notion of the work and the organisation.

3.1.2.2 Field Studies

To fully understand how a work actually is conducted, discussions in a seminar series are not sufficient. Field studies including observation of workers are an important prerequisite to design of support systems. Without field studies uninformed systems designers may draw attention to unimportant issues, draw false conclusions, and lead the work in an inappropriate direction. Knowledge must be obtained from the very place where the IT-system is to be implemented and used.

In parallel with the seminar series, two different workplaces represented in the seminar group were studied in more detail. Two one-day field studies were conducted to study the work in practice and to find additional important details in the work practice. Participatory observations were mixed with short interviews during the day.

3.1.3 Vision Seminar Results

Eventually, the work in the vision seminar resulted in a number of suggestions for improvement in order to achieve a good and healthy future work. We worked with the result in terms of aspects on the future work. These aspects are very specific for the work within the elder care in Kortedala, and some of them quite specific for the respective work place for each participant in the seminar group. Here follows a selection of the more general and concrete aspects on the proposed, future work.

3.1.3.1 Autonomous Groups

The groups carrying out the practical health care work should be autonomous and be given a broader, local, responsibility than today. They are to organise and carry out their own planning, dispose their own resources and undertake their own quality assurance. As discussed earlier, it is important that increased responsibility also is supported by increased possibilities.

It is our conclusion that it is locally, on the operative level where the work is actually performed, the best and most effective planning can be accomplish. It is also there the best priorities can be made. It is the working teams that possess the valuable knowledge of the actual conditions. It is down there, "on the floor", on site in the care takers home, where quality improvements can be achieved.

3.1.3.2 Coherent Health Care Planning

We want to achieve a common use within the organisation of the individual care plans used in the elder health care. The whole chain of care planning; from the very first decision of assistance, care planning, intervention planning, execution and documentation and finally follow-up and evaluation – all shall be well planed, well known and concurrent throughout the whole organisation These objectives already exist today, but they are poorly fulfilled. It is desirable that the health care personnel participate in the decisions of assistance to assure that the decisions become practical.

3.1.3.3 Increased Economic Responsibility

The economical responsibility shall be moved further down in the organisation, down to where the costs appear. Providing the right means in terms of access to information, education, authority and competence, it is where the cost actually occurs that they can be controlled. To facilitate a more effective use of the resources, it shall be possible to make economic plans in advance, not just to be able to look backwards to se previous expenses.

3.1.3.4 Increased Professionalism

Occupations in the elder health care deserve a better reputation. The feeling of being branded as a maid is frequently encountered. We want to work for a more pronounced professionalism where the personnel really are allowed to perform the undertakings they are educated for. This is also facilitated by the earlier discussed aspects of autonomous groups and increased economic responsibility.

3.1.3.5 A Healthy Work

By a strong focus on the work environment, a number of aspects can be improved and a healthy work obtained. A well established staff welfare and influence on the organisation will decrease the sick leave and reduce the employee turnover. Consequently, this will attract new personnel and provide for a higher quality in the entire health care sector.

3.1.4 Supporting the Future Work with IT

Supporting IT systems can not by themselves fulfill an organisation's goals, but they can hopefully contribute by being a powerful support, and result in a positive development of work and services. The technology can be used as a driving force in development of the work. What we have found is that through the used participatory process, we have been able to develop new efficient work processes that not can be achieved without the new supportive technology.

To implement the new work procedures, effective information access and communication means is a necessity. A supporting IT system can, properly designed, meet these requirements and facilitate the future, proposed work. Here follows a selection of how technology can facilitate the requested changes of today's work described in terms of aspects in the previous chapter.

The proposed *autonomous groups*, carrying out the actual outside service, are to be given access to the information needed, on the very place where the work is conducted. This requires a mobile IT system providing access to e.g. schedules for both personnel and care takers, where planned and accomplished efforts are accessible. Further, medical information like patients medicine lists and general medical information are to be part of the accessible information.

The chain of care planning, essential for the *coherent health care planning*, shall be supported by an IT system accessible by all parts in the care chain. A well designed IT support provides easy information access.

Proper information access also provides for the proposed *increased economic responsibility*. Responsibility must be followed by authorization. A supporting IT system can provide necessary economic information, e.g. costs for services, materials and time as well as available resources and possibilities. Through that, home health care teams can be given larger economic responsibility followed by better possibilities to provide better care services. The most efficient economic control occurs when it is transformed into a real-time planning of the available resources. To achieve this, the economic planning, control and evaluation must be made where the costs occur, i.e. out in the operational teams.

3.2 Design of Future IT Support

The results above indicate where in the organisation and in the work practice improvements can be made, and where and how a supporting IT system can have a positive effect. The detailed functionality and design is not specified within the project. However, a knowledge base is established and first prototype for the design has been developed.

It is important to fully understand that a professional user's task or goal is to perform a qualitative work, not to execute a set of operations on a computer. Users should be allowed to keep focus on the actual work and not bother how to manage the IT system.

Knowledge about users' cognitive abilities can be used to understand important requirements for design of functionality and interaction [9]. Examples of important design requirements are:

- Users must be allowed to have a complete and undisturbed focus on the work tasks, and handling of the technical support systems must be more or less automated.
- The support system must provide appropriate functionality (have a high utility).
- The interface must present information in a way which is consistent with human perceptive and cognitive skills and limitations, e.g. concerning limitations in short term memory.
- The interface must provide accurate feed-back, so that the user can understand it and is in full control of what happens.
- The user must always be in total control of the work process and the support system must never take over control. This is important in order to avoid a feeling of being controlled by the system.

3.2.1 From Description of Work to IT System

To specify the specific work tasks, activities, information content, communication, need of assistance and tools, we have used scenarios describing the work. Based on larger scenarios we created more narrowed and detailed scenarios describing the main work tasks that constitute the work in Kortedala home health care.

Based on each scenario we identified specific work tasks. There is no clear division of work into work tasks, but we tried to find a definition that provided a good structure for the work. Each work task consists of a number of activities, all with a need for information and communication. Inversely, a sequence of connected activities with the purpose to achieve a certain goal can be considered as a work task.

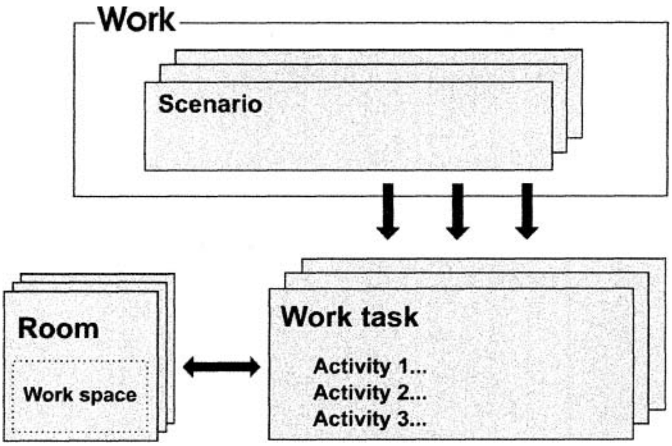


Fig. 3. Based on a number of scenarios, important work tasks can be identified. One work task consists of a number of activities and can be supported by a room or work space in the IT system's interface

3.2.2 Design for Mobile Work

Mobile technology can be used in order to facilitate for a stationary work to become mobile. It can also be used to support an already mobile work, as in the case of the VIHO project. The work within the home health care for elderly is, and has always been mobile, but has for long lacked IT support for the mobile work. The work place where the use of a mobile IT system occurs is often in the care takers' home or some way on the road between care takers and the office. In the VIHO project we have carefully striven for solutions that consider mobility characteristics in the work. During the work in the seminar series, three important qualities of mobile IT system have been recognized:

3.2.2.1 Quick and Effective

In the home health care, one of the mobile IT systems' most important attributes to consider is to be fast. The time available to start a device, enter or read information etc. is strongly limited. Everything that is not perceived as simple, quick and supportive will be considered as bothersome and will hardly be accepted and used by the personnel.

Examples indicates that long start-up time results in that users do not bother to read or write care documentation as intended. In the case described in [4], the start-up time includes booting the device, establish the appropriate network connections , start the application – everything that is needed to set the device in such a state that the user actual can start working. This requires good hardware performance with fast networks and connections using effective verification and security procedures. Applications and interfaces are to be usable and quickly let the user carry out the actual work with a minimum of navigation in the interface.

3.2.2.2 Durable and Reliable

In distinction to desktop work, using a stationary IT system, the circumferential environment and the contextual aspects are of much grater importance when using a mobile IT system. A mobile work imply for a constantly changing environment includes unpredictable events.

A user of a mobile device and IT system is often in need of carry out something else at the same time as using the device; driving a car or traveling with an underground railway, giving medicine, bandaging a varicose ulcer or writing a prescription. The environment can be very noisy as well as silent and include water, vomits, blood, coffee and hasty movements. Much can happen that interrupts the use. A good mobile IT system for home health care shall, as far as possible, manage the unexpected situations that can occur in the variable environments where it is used.

Fig. 4. Nurse in the home health care receives an urgent phone call on her way from the car to a patient

3.2.2.3 Flexible

Given the nature of mobile technology, it has to be small and easy to carry along. This affects the technology in a number of ways. When physical size is reduced, the technical performance is deteriorated. Smaller CPUs are getting slower, memory and battery time is impaired as well. As another consequence of the reduced size, the probably most important interaction channel, the screen, is reduced in size. A smaller screen implies bigger challenges to design usable graphical interfaces to the system.

Design knowledge becomes very important and must be carefully adapted to the conditions that exist for use in mobile contexts.

3.2.3 The VIHO Prototype

This very first initiated phase of practical design constitutes the end of the VIHO project. We do not intend to produce a complete designed interface, but we have provided a basis for future design work.

The device of our choice does not yet exist. Technology is by all means available, but we have not found a device on the market that fully lived up to the requirements specified in the seminar work. We preferred a light weight tablet PC in the size of half an A4 paper. Not to big, clumsy and circumstantial to carry along but still large enough to provide a large readable screen able to display much information at the same time. Our early proposal of the interface of a supporting IT system for the mobile home care services was prototyped and look like this:

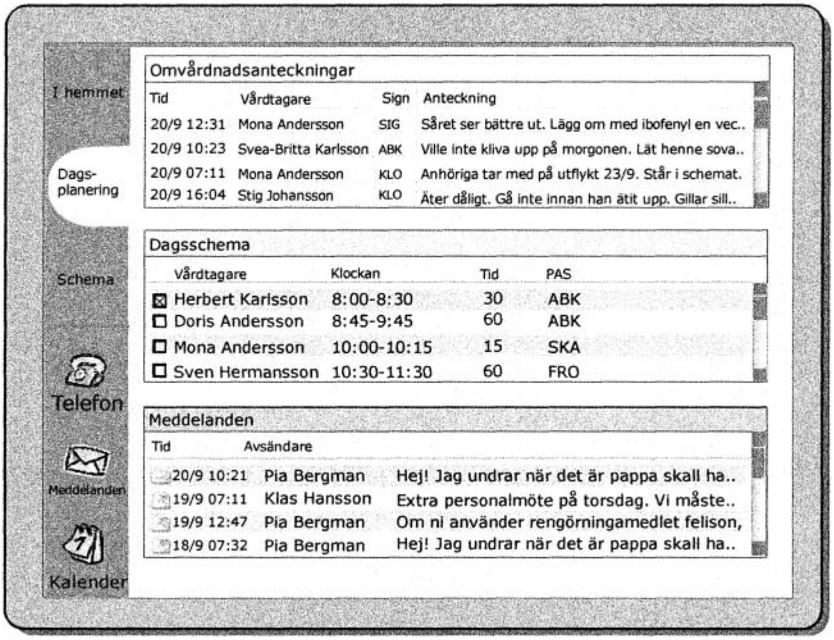


Fig. 5. Early prototype, showing overview workspace

Figure 5 illustrates when the user is using the day planning tool, showing an overview work space. This is the planning tool to use in planning situations, preferably when starting the working day. Here the user can get a quick view of the present status of the work situation. Information is displayed about today's care takers and the scheduled visits, recent documentation and messages. Choosing a specific patient in the application displays more detailed information.

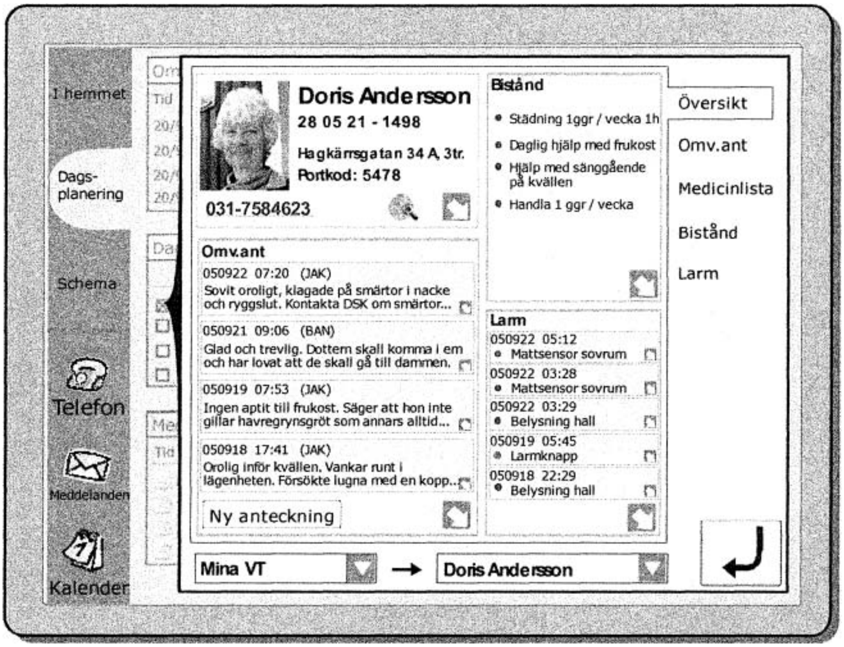


Fig. 6. Early prototype, showing detailed view

Through design of the proposed application, it is our intention to support the proposed new work by paying attention to the aspects produced in the seminars and described in the scenarios. In this case the aspects about autonomous groups, increased economic responsibility and coherent health care planning are specifically concerned. The application clearly informs about how much assistance a patient is entitled to and the nature of the assistance. This assistance is a result of one of the first decisions in the chain of care planning. If the practitioner, the user of the IT-system, finds this amount of assistance indefensible when carrying out the assistance in practice, there are routines for sending this feedback back in the decision chain to the decision-makers. In this way the feedback forms the basis for and triggers a new care planning decision.

In this way the personnel are also given a larger economical responsibility when carrying out the work on the field. They are provided with the recent information by the IT system about their patients entitled care, but are permitted to make individually decisions effecting the present situation. Sometime a patient can need more attention at the expense of some others. This is a delicate balance and has to be handled by the personnel in the field, given supporting recourses and responsibility.

3.2.4 Summary

The methodology used is intended to support a logical chain from development of work processes and work organisation, via specification of new work processes, tasks and activities to specification of requirements for supportive technical systems and design of appropriate and efficient user interfaces. Main links in the chain are:

- Analysis of present work processes and organisation. Identification of problems and development.
- Specification of important aspects that must characterise the new future work processes and organisation, i.e. the 'aspect list'.
- More detailed specification of the new work situations in terms of scenarios.
- Identification of tasks and activities of the new work situations.
- Definition of a set of 'rooms' for the technical support system, where each room corresponds to one work task for one user.
- Specification of information elements and tools needed in each specific room.
- Detailed design of the user interface in each specified room

4 Discussion

When introducing new IT-support systems in a work environment, it is important to keep in mind that the only relevant rationale is to contribute to improvements of the total work situation. This means that the new technical support system, to be justified, should contribute to e.g. improved quality, efficiency and a healthy and sustainable work. To improve the quality and the efficiency, it is required that not only the technology is developed, but that also the redesign of organisation, management, work processes etc must be considered. A healthy work requires that the user can handle the system with high usability and that he or she is in total control of the work process. This requires that the functionality and the dynamic interaction between the system and the user are well adapted to local needs, requirements and expectations. In order to achieve this, a user centred development process is needed. We have found that the use of workshops, or seminars, with representative groups of users, who are allowed to spend enough time to specify requirements and scenarios of future work processes, is an efficient model.

In the VIHO project, we have applied an action research approach. The seminar group consisting of researchers and practitioners has been a valuable source to knowledge about how the work presently is carried out and how visions and scenarios of the future work can be formulated. There are some problems related to this methodology, which are important to understand and consider. The seminar group must consist of experienced and engaged professionals, and they must also be given enough time to live up to the expectations. It is also important to have a very clear understanding about the mandate of the seminar group. They must be allowed to act independently at the same time they must deliver a result that is realistic. As the result is a vision of future work processes and technical support systems, it is not sure that the proposed solutions ever will be implemented, at least not to all details. This can cause false expectations and frustration.

5 References

1. Olsson, E., Johansson, N., Gulliksen, J. and Sandblad, B. (2005) A participatory process supporting design of future work, Technical report, Department of Information Technology. 2005-018. ISSN 1404-3203.
2. Åborg, C. (2002): How does IT feel @ work? And how to make IT better. Computer use, stress and health in office work. Doctoral thesis, Uppsala University, Sweden.
3. Karasek, R. and Theorell, T. (1990) Healthy work: Stress, productivity and the reconstruction of working life, Basic Books, New York, USA.
4. Verburg, R., Testa, S., Hyrkkanen, U. and Johansson, N. (2006) Case Descriptions of Mobile Virtual Work in Practice, In: Andriessen, E. and Vartainen, M. (Eds) Mobile Virtual Work - A New Paradigm? Springer-Verlag.
5. Masters, J. (1995) The history of Action Research. In: Hughes (ed.) Action Research Electronic Reader. The University of Sidney. Online:http://www.scu.edu.au/schools/gcm/ar/arr/arow/default.html (download date 2005-04-01).
6. Göransson, B. (2004) User-Centred Systems Design: Designing Usable Interactive Systems in Practice. Ph.D. dissertation. Acta Universitatis Uppsaliensis. Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 981. Uppsala, Sweden.
7. Schuler, D. and Namioka, A. (Eds) (1993) Participatory Design – Principles and Practices. Lawrence Erlbaum Associates, New Jersey.
8. Greenbaum, J. and Kyng, M. (Eds) (1992) Design at work: cooperative design of computer systems. Lawrence Erlbaum Associates, Inc.
9. Sandblad, B., Lind, M. and Nygren, E.(1991) Kognitiva arbetsmiljöproblem och gränssnittsdesign. CMD-rapport nr 20/91. Uppsala University, Sweden.
10. Scandurra, I., Hägglund, M., Johansson, N., Sandblad, B. and Koch, S. (2003) User needs for development of context dependent devices in mobile home care. In: Chittaro L (ed), Human-Computer Interaction with Mobile Devices and Services, LNCS 2795, Springer-Verlag, pp. 445-450.
11. Henderson, A., and Card, S. (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) pp 211-243.
12. Gulliksen, J., Lif, M., Lind, M., Nygren, E., and Sandblad, B. (1997) Analysis of Information Utilisation. International Journal of Human-Computer Interaction, 9 (3).
13. Lif, M., Olsson, E., Gulliksen, J. and Sandblad, B. (2001) Workspaces enhance efficiency - theories, concepts and a case study, Information Technology & People 14 (3) 261-272.
14. Preece, J. (1994) Human-Computer Interaction, Addison-Wesley, Wokingham, England.

Design of a Resource Allocation Planning System

Kelly O'Hargan and Stephanie Guerlain

University of Virginia, Charlottesville, VA, USA
guerlain@virginia.edu

Abstract.

This paper proposes a generic human-computer software user interface design, called the Resource Allocation Planning System (RAPS), designed to support a person making resource allocation decisions. Although there are many algorithms for automatically solving resource allocation problems, it is often the case that human judgment is also required. Also, while there are software user interfaces to support decision-making for specific resource allocation problems, most of them serve more as organizational charts than as decision-support systems, and most of them become increasingly difficult to use as the size of the resource allocation problem increases. This paper discusses the design and rationale for RAPS and gives an example of how RAPS can be adapted to a specific resource allocation problem.

1 Rationale and Objectives

Resource allocation is a common type of task that requires complex cognitive reasoning; it requires assigning assets to demands given a set of constraints. A decision-support system (DSS) can reduce cognitive demands by organizing information effectively and making aspects of a task that are always true automatically executable, therefore allowing a decision maker to focus cognitive resources on those aspects of the task that require human input and judgment. For example, when a user is trying to make a decision based on a set of options, the cognitive workload on the decision-maker can be reduced by limiting the options presented to only those options that are feasible. Furthermore, the difficulty of a cognitive task can be decreased (or increased) depending on the tools and representations available to the decision maker for tracking information and task completion. Effective representations provide decision makers with an "external memory" (e.g., information is represented directly "in the world"), thereby eliminating the need to remember that information "in the head". Ineffective representations, on the other hand, require the operator to remember important constraints, relations, procedures, etc. "in the head' while working on a problem, adding cognitive burden and increasing the likelihood of errors. This is known as the Representational Effect [1].

Please use the following format when citing this chapter:

O'Hargan, K., Guerlain, S., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 67–92.

Complex domains with a large set of tasks and task situations that must be managed have the propensity to increase in complexity to a point that a single representation cannot reasonably embody all of the necessary information (relationships, details, constraints, task progress, etc.). Therefore, a complete user interface design must include a set of corresponding representations and offer the functionality for a user to easily switch from one task to another, while maintaining situational awareness of the complete system. This is known as Workspace Navigation [2].

The above discussion now frames the research described in this paper: the creation of a Resource Allocation Planning System (RAPS), a user interface and associated information organization functions for medium to large-scale (25 to 100+ entities) resource allocation problems The intended benefit of RAPS is to provide the user interface design community with a clear methodology for how to design resource allocation information displays, along with associated information organization functions, to enable a human decision maker to be included in the decision making process and make informed judgments without becoming overwhelmed. The possible applications of RAPS are extensive. The design is specified such that it can be easily adapted to fit generic resource allocation problems. As a demonstration of its applicability, this paper gives an example of specific instantiation of RAPS.

2 Resource Allocation

According to the Encarta dictionary, resource allocation is "the activity of deciding how resources such as money, assets, and personnel should be used in order to achieve a particular aim." [3] This means that a decision-maker has a set of resources and goals, and the problem is determining the best way to assign those resources to meet those goals. A common set of characteristics for resource allocation problems are shown in Table 1.

In addition to the constraints defined by the generic requirements of the system, there may also be run-time constraints associated with the problem. These constraints may be created by: new information the decision-maker receives, known information that is not included within the model, special circumstances, or changes to the system. Run-time constraints can also simply be the unconscious opinions that affect a decision-maker's judgment calls. Take the example of someone assigning shifts for a group of nurses at a hospital. The resources are the nurses and the demands are the shifts that need to have nurses assigned to them, (see Table 2). If run-time constraints are not met, perhaps because the scheduling system does not take personal preference into account, then the nurses will receive the schedule and switch between themselves until everyone is happy with the shifts they are covering. However, if the scheduler had been able to take all of the individual's preferences into account, then there would be no need for swapping shifts.

Table 1. Characteristics of Resource Allocation Problems

Characteristic	**Definition**	**Examples**
Resources	Assets available to be distributed	time, money, personnel, goods
Demands	Needs that must be met	tasks, jobs, areas, time-slots
Hard Constraints	Specific requirements associated with classes of resources, classes of demands, or resource/demand combinations	all resources assigned to demand X must have characteristic Y
Soft Constraints	Desires or costs associated with classes of resources, classes of demands, or resource/demand combinations	resource X is best for demand Y, and has a lower cost associated with it, since it has characteristic A
Run-Time Constraints	Hard or soft constraints that 'pop up" or are not generic to classes of resources and constraints	Resource X1 requires demand Y2 due to outside information not known ahead of time
Structure	Whether one asset can be assigned just once or multiple times, whether one demand requires just one or multiple assets	- Resource A can only be used once (missiles assigned to targets) - Resource B can be scheduled for several different time slots (competitor entering many events) - Demand C requires multiple assets (project teams need 4 people)

Resource allocation is a broad definition that encompasses a large set of different types of problems. Resource allocation can refer to how a business is going to spend its money, how a lawyer is going to allocate his or her time, or how the military is going to distribute its manpower. The different types of resource allocation problems are too many to name, but some of the most common and basic problems are

characterized in Table 3. Many resource allocation problems could have a different form, or are a combination of these different variations.

Resource allocation can be broken down into a set of steps: 1) generating potential solutions, 2) comparing alternate solutions, and 3) selecting and implementing a particular solution. Along the way, the decision maker might want the option to be able to understand the costs and benefits associated with the options available and to adjust solutions as desired to ensure that all of the requirements of the system are being met to the extent possible before selecting a solution.

One of the main goals of this research is to design a flexible, or "adaptive" user interface to support the decision-making process of a user or team of people faced

Table 2. Characteristics of Resource Allocation of Nursing Shift

Characteristic	Nursing Shifts Example
Resources	Nurses
Demands	Shifts at a hospital
Hard Constraints	-each shift is covered by X nurses -each nurse receives Y-Z hours -no nurse works more than A hours in a 24 hour period
Soft Constraints	-no nurse works more than X hours in a week -each nurse gets Z hours between shifts
Run-Time Constraints	-nurse X can only work nights -nurse Y doesn't work on Sundays

with a resource allocation problem. A Decision Support System (DSS) manages the data essential for decision-making and translates the data into information that once presented to the decision-maker provides the information needed to make accurate decisions [4]. There are two main kinds of DSS: cognitive prostheses, in which the computer makes a decision that the user has to approve; and cognitive tools, in which the user is supported in his or her own decision making process by a comprehensive display of all of the essential information [5].

Cognitive Prostheses put the human in a subservient role by having the computer automatically make decisions that are then interpreted and approved by the user [5]. However, if a user developed his or her own solution to the problem, there is rarely any way for him or her to effectively compare his or her answer with the one found by the model [6]. Thus, the user is restricted to the solution set defined by the algorithm. Also, since the algorithm is based on numeric values that represent different aspects of the problem space, it is often difficult to see the logical progression of the model and how particular solutions are determined. This makes it hard for a decision-maker to understand the factors driving a particular solution to the problem or how adjustments may affect the solution. Similarly, using an algorithm as the chief decision-maker limits the user's ability to include additional user-defined, "run-time" constraints. User defined constraints can be important since the user may have additional information that was not available when the model was built, he or she may want to adjust the weights associated with the importance of different factors, or he or she may simply want to include his or her opinion in the decision-making process. Often resource allocation problems will have a number of "close" solutions where all of the demands are met and the mathematically calculated cost is the same for a number of different options. This is a case where human-judgment can be used to choose the best solution, if the DSS supported that functionality.

Table 3. Typical Resource Allocation Problem Classifications

Classification	Resources	Demands	Hard Constraints	Soft Constraints	Structure
Temporal	assests that need to be scheduled	times during which demands need to be met	all resources are scheduled for all of the times needed, if possible only one resource scheduled at a time	certain resources must be assigned to particular time-slots certain time-slots must be met by particular resources	time-slots with at most one demand assigned to each
Spatial	assests that are in a certain location	locations where demands need to be met	the minimal amounts of required resources are allocated to designated areas	the consequences of certain areas not receiving the resources they need	multiple resources assigned to multiple areas
Dynamic	assets that move over time	locations where demands need to be met at particular times	a particular set of resource characteristics are needed in order to satisfy a demand	there are different costs associated with different resource/demand assignment combinations	particular resources assigned to particular demands based on efficiency of resource/demand combinations

Due to the limitations of the Cognitive Prosthesis approach, effort has been focused on providing decision-support through Cognitive Tools approach. Cognitive tools are decision-aids that have been developed to provide support to decision-makers by presenting information to the decision-maker in an effective way and allowing users to track their own decision making progress, often using information visualization techniques.

Information Visualization uses graphical means such as information placement, size and color to communicate ideas and information and to relate and compare entities of a system [7]. The purpose of information visualization is to represent relationships between entities comprehensively, and use visual representations that model the physical system in a way that is intuitive to the user. Effective information visualization allows a user to understand how different aspects of a system are related, simply by how the information is presented.

3 Information Visualization for Resource Allocation Problems

There are a number of visualization methods that have been developed for different types of resource allocation problems, such as: matrix displays, time-line displays, spatial displays, and combination displays. The basic idea of a matrix is for the options to represent rows down the left side, and the columns to represent either particular characteristics or items the options can be applied to meet. The cells within the matrix represent the relationships between the intersecting rows and columns. A matrix is particularly useful for multiple objective functions since the result of each possible row/column combination is summarized in the intersecting cell of the matrix. Unfortunately, in a matrix design, the visual representation increases in size as more entities need to be represented, so the user is forced to scroll through multiple windows or page through multiple screens to find the information he or she needs [8]. One approach for combating this problem is to use the Table Lens approach [9], which allows a user to visualize a larger table than could normally be displayed by expanding the cells of interest, while minimizing the size of the remaining cells. This type of interface permits the user to view the detailed information he or she is considering, while maintaining situational awareness of the complete system.

Temporal resource allocation problems are generally scheduling problems. Information visualizations for scheduling problems are typically set up like calendars with the days of the week listed along the top with more specific times listed down the left side, see Figure 1; or time-slots listed along the top and ways to fill the time-slots listed down the left side, see Figure 2. Within each time-slot for each day, the information is presented within the cell, indicating what will be happening during that time period, i.e. who will be working, what will be on TV, who will have an appointment, etc. The example of scheduling nurses for hospital shifts is a temporal resource allocation problem.

	Monday	Tuesday	Wednesday	Thursday	Friday
9:00 - 2:00	names	names	names	names	names
12:00 - 4:00	names	names	names	names	names
2:00 - 7:00	names	names	names	names	names

Fig. 4. Weekly Schedule Example

Channel	7:00	7:30	8:00
USA	Law and Order		Law and...
TNT	Action Movie		
Fox	News	Simpsons	Simpsons

Fig. 5. Time-Slot Example

For temporal resource allocation problems it is common for things, such as shifts, tasks, appointments, etc. to overlap. When items overlap the matrix becomes more graphical (and potentially dynamic if things are changing over time) to enable seeing those overlaps.

If the characteristics and decision-making criteria of a system are geo-spatially oriented, then it is logical to organize the information onto a map. A map orientation allows the decision-maker to visually interpret the spatial relationships within the system.

For resource allocation problems that have temporal and spatial aspects, one might see a combination system that represents this information using all of the features just described.

4 Designing a Generic Resource Allocation Interface

The Resource Allocation Planning System (RAPS) is designed to be a generic user interface to allow a person to visualize resource allocation problems and support a user in reallocation situations and optimization problems, especially for medium to large-scale (25 to 100+ entity) environments. The main RAPS interface is set up as a matrix that automatically adjusts in scale to allow the user to view the entire system at all times.

The table is set up with the resources listed as the row headings down the left side, and the demands listed as column headings along the top (Figure 3). The colors of the demand and resource title blocks indicate their priority or importance to the user: green indicates lower priority/importance followed by orange, dark red, and then bright red. Depending on the particular system, the color scheme can be adjusted to accommodate additional levels of priority and to indicate areas that need attention. The cells within the table give the user information about how the resources and demands interact. A colored (non-black) cell indicates that the resource of that row satisfies the hard constraints of that column's demand and that that resource could be reallocated to meet that column's demand. The color of the cells indicates additional information about how that row's resource can meet that column's demand. In this example, none of the demands of the system are currently in need, so all of the colored boxes are gray. Black cells indicate that the resource of that row cannot be used for that column's demand due to not meeting that demand's currently specified constraints. The text within a cell indicates that the resource from that row is currently being assigned to meet the demand of that column. The text generally indicates additional information about that assignment, for instance when that resource will meet that demand or how much it will cost for that resource to meet that demand.

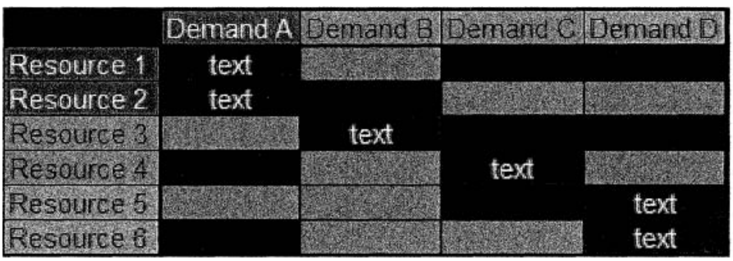


Fig. 6. Decision-Support Tool

If a demand is in need of an additional resource then the system indicates this by changing the color of the gray boxes to a color associated with that demand's importance. In the example shown in Figure 3 none of the demands of the system are currently in need, so all of the colored boxes within the matrix are gray. If it was determined that an additional resource was needed for Demand C, which is an orange, or medium level demand, then the gray cells indicating resources that could meet that demand would turn orange, (see Figure 4). Using this example, the user has two options for resources that could meet Demand C: resources 2 and 6. Resource 4 is already allocated to Demand C, but Demand C requires an additional resource at this time. When making this reallocation decision, the user wants to choose the resource with the least amount of cost associated with it. In this case, Resource 2 is currently meeting Demand A, which is of high importance, while Resource 6 is currently meeting Demand D, which is of low importance. If the user desires, he or she can retrieve additional information about either the resources or the demands involved. If Resource 6 is reallocated to meet Demand C, the decision-support tool will adjust to show the needs of Demand C being met and the needs of Demand D not being met, since Demand D has lost the incorporation of Resource 6 (Figure 5).

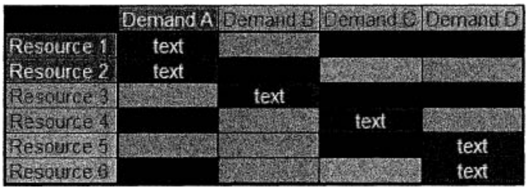


Fig. 7. Demand C in Need

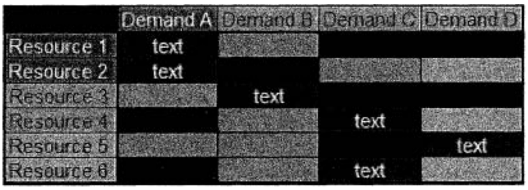


Fig. 8. Demand D in Need

4.1 Selection Function

One of the main benefits of RAPS is allowing the user to visualize the entire state of the system at once, while being able to quickly retrieve necessary information and quickly determine if there are areas of the system that need attention. In Figure 6, 38 demands and 58 resources are displayed, and even with all of this information the user can quickly see what areas of the system need attention. In this example, there are two demands that need resources: one demand of medium (orange) importance, and one of high (dark red) importance.

As the user seeks to address these demands, the display adjusts to provide the user with all of the necessary information. When a user selects a demand, the column expands, as well as the rows of the resources that could be reallocated to it (Figure 7). Similarly, when the user selects a resource the row expands, as well as the columns of the demands that the resource could meet (Figure 8).

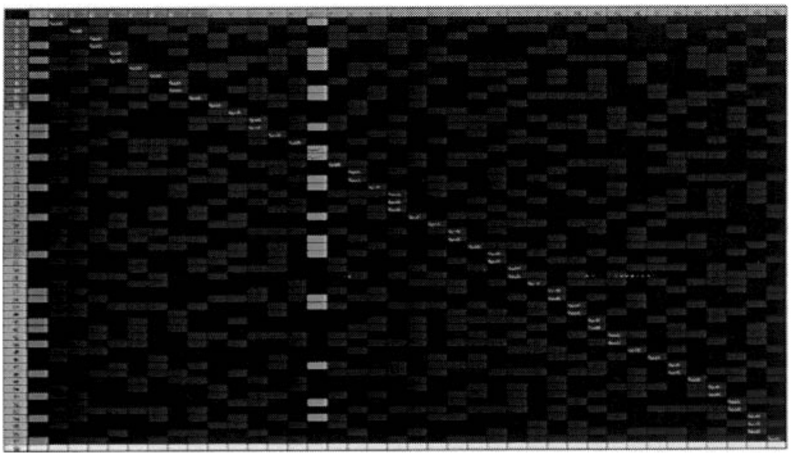


Fig. 9. Basic Overview with Unmet Demands

It is the intention of this feature for the display to adjust such that the information text displayed in the cells would be easily legible to the user, allowing him or her to make informed decisions. A search function is included within the system, to assist the user in selecting particular entities.

4.2 Information Windows

Information Windows are used to provide the detailed information concerning a particular demand or resource and how it is related to the rest of the system.

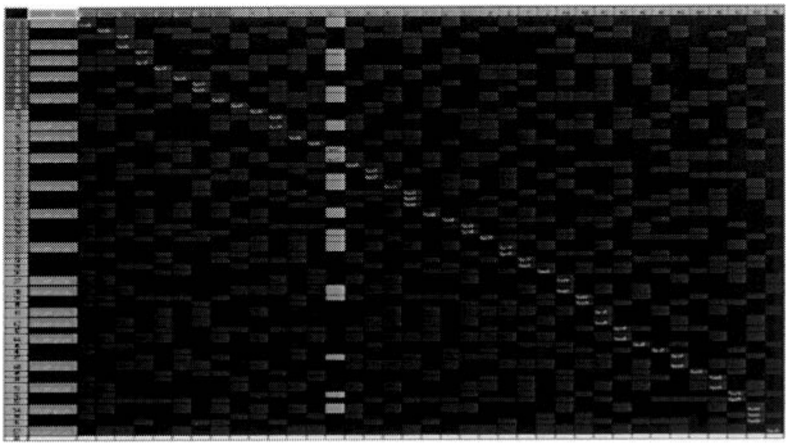


Fig. 10. Basic Overview with Demand A Selected

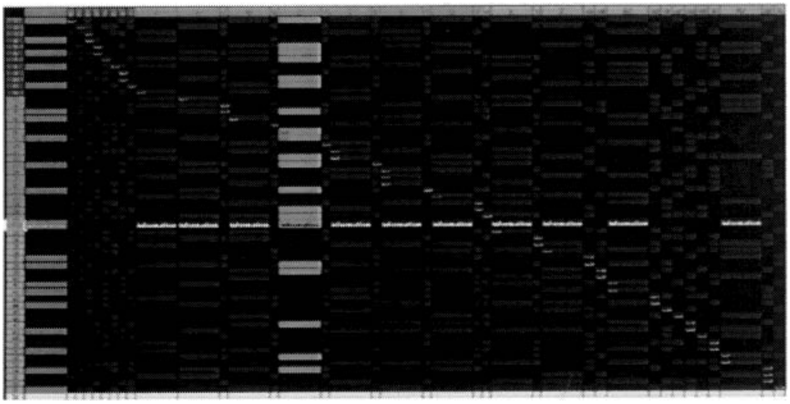


Fig. 11. Basic Overview with Resource 32 Selected

4.3 Resource Information Window

The Resource Information Window provides the user with all of the vital information related to a particular resource (Figure 9). The window displays all the important information about the resource: the name of the resource, a description of the resource, all of the important general characteristics about the resource, any additional information the user may want to be aware of, where the resource is currently allocated, and where the resource could be allocated, including information concerning how the resource would meet the desired characteristics of those demands. The list includes all of the possible demands the resource could be reallocated to, presented in a ranked order, with the demands in greatest need and the demands that the resource could most effectively meet listed at the top. In order for a demand to be listed, the resource must be able to meet the "hard constraints" of the demand. The ranking in the list is an indication of how well the resource would meet the "soft constraints" or desired characteristics of the demands.

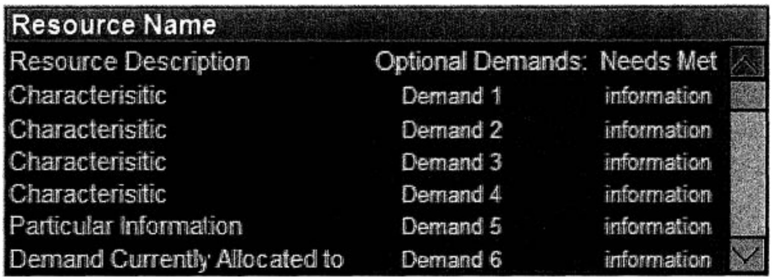


Fig. 12. Resource Information Window

4.4 Demand Information Window

The Demand Information Window is similar to the Resource Information Window, except the information is related to the demand rather than the resource in question (Figure 10). The Demand Information Window includes: the demand's name, the demand's description, the general characteristics about the demand and any additional information which may be important to the user, the resource(s) currently allocated to it, whether it needs an additional resource, and a list of the optional resources that could meet the requirements of that demand. Again, the resources listed are those that can meet the "hard constraints" of the demand, and the order of the resources indicates how well the resources meet the desired characteristics for that demand.

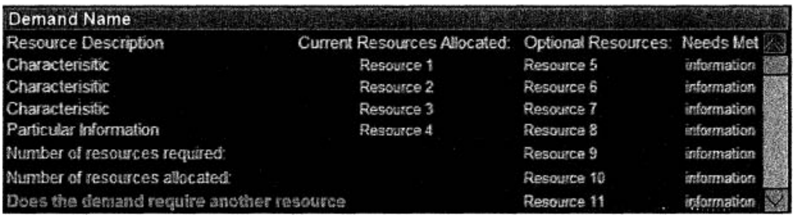


Fig. 13. Demand Information Window

4.5 Filtering and Sorting

In addition to displaying all of the information necessary to monitor the system, RAPS has a number of sorting methods to assist the users in organizing the information to best suit their desires and/or needs. The organization methods vary between different systems depending on the characteristics associated with different resources. For example, typical organization methods include: by priority/importance (either high-to-low or low-to-high), by name, by number, or by areas that need the most attention. The different organization methods are typically included in a tool-bar, and the current organization method is highlighted. In the example shown in Figure 11, the attention method (depicted by a yellow exclamation point sign), of organization is being used, instead of organizing the information by high-to-low importance or an element of time. In all of these organization methods, the demands are re-ordered from left to right and then the resources currently assigned to them reorganize to line up with the demands in a diagonal manner, see Figures 6-8, which are in a high-to-low priority organization. If a user makes a change to the assignments, the display will adjust to indicate the change, but it will not reorganize. A user can reorganize RAPS or switch organization methods by clicking on the organization method of choice from the toolbar.

RAPS also includes multiple filtering options to hide information relating to resources or demands that the user cannot affect or does not want to change, in order to reduce the number of entities viewed by the user. Filtering options can vary greatly between the different systems due to the range of characteristics that different systems would filter by. If there is a characteristic that some demands possess that requires an associated characteristic of the resources, then those demands and resources that do not contain those characteristics could be filtered out. The filtering options are designed to remove unnecessary information from the display to limit the options presented in a decision-making situation. For example, for the two filtering options shown in Figure 12, the filtering option with the number sign "#" and the number 4 is used to filter how many levels of preference are presented to the user. The check mark is used to minimize information from resources and demands that have already been effectively assigned.

Fig. 11. Sorting Example

Fig. 12. Filtering Example

4.6 RAPS Optimization Tools

Another helpful aspect of RAPS is the Multiple Reallocation De-confliction Feature, which allows the user to select a group of demands that still need resources and receive a Multiple Reallocation De-confliction Window. The Multiple Reallocation De-confliction Window is an organized table of the demands, and the best set of resources to meet those demands (Figure 13). The user is presented with the algorithm defined "best options" and all of the characteristics of those options which could influence how those options could best be allocated to meet the needs of the demands. Generally the name cells for the resources and demands are color-coded to indicate their importance. Also, the indication of fit is generally color-coded to help the user visually make reassignment decisions. Using this tool it is easier for the user to choose between a set of resources and demands and how they should be assigned to each other. The user can select the cells of the intersecting resources and

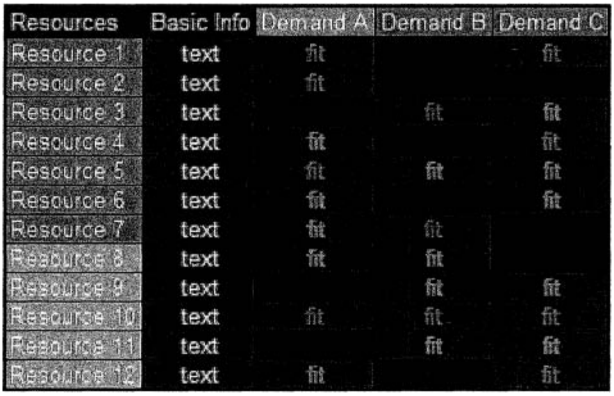


Fig. 14. Multiple Reallocation De-confliction Window

demands, and then, after all of the demands have the needed resources allocated to them, the user can reassign all of the selected resources to the indicated demands at the same time.

4.7 RAPS Special Circumstances Tool

Often in decision-making problems there are additional characteristics that should be taken into account that do not fit into the predetermined factors of the system. There may be a characteristic that only affects a particular resource or demand, so to include it in a decision-making algorithm and to appropriately weight the importance of this characteristic can be difficult. This is a problem that RAPS is particularly able to address, since RAPS allows a person to be within the decision-making loop, able to make decisions regarding factors that do not have particular weights assigned to them.

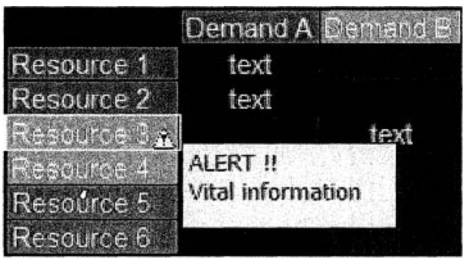


Fig. 15. Unique Information

This additional information is generally presented through written statements included within the information windows and is also indicated by an alert note that appears in the corner of the resource or demand name cell, indicating to the user that there is important information he or she needs to know about this particular entity (Figure 14). Users can also create their own alert notes to help them remember why they have made a particular decision, or to prevent themselves from making a particular decision.

4.8 RAPS State of the System Indicator

There are several ways RAPS is designed to indicate the state of the system. Depending on the characteristics of the system, whether it is acceptable for certain demands not to be met or certain resources not to be used, the system will use either a system summary to alert the user to areas that need attention (Figure 15), or include an additional row and column to the matrix which indicates to the user when a particular resource is not assigned or when a demand is not being met (Figure 16).

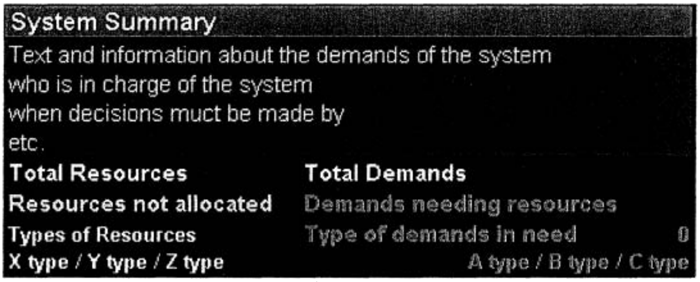


Fig. 16. System Summary Example

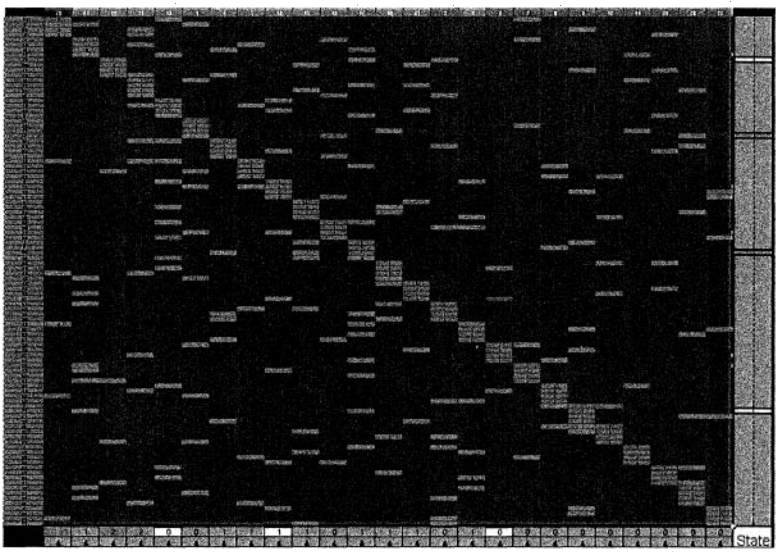


Fig. 17. Row/Column Overview Example

5 Application of RAPS

An example of the applications of RAPS is discussed next. The example is designed to assist the University of Virginia Department of Systems and Information Engineering when assigning 4th year engineering undergraduate students to "capstone" research project teams. Students rank projects in order, based on their preference of the projects they would like to work on, but the number of students on each project needs to be distributed evenly, so it is necessary to sometimes "add" students to less popular projects, preferably removing them from teams that are too large, while not moving a student to a project that is ranked below his or her 3rd or 4th choice project.

5.1 The Capstone (C-RAPS) Design

Each year all of the (70-100) 4th year Systems Engineering students at the University of Virginia are assigned to Capstone projects. Each of the students ranks all of the (15-30) Capstone projects available from their first to their last choice, then all of the rankings are organized into a min-cost flow problem. In this example, the Capstones are the demands and the students are the resources to be assigned. The hard-constraints for the assignments are that each student is only assigned to one project and that each Capstone project has a certain number of students assigned to it. The soft constraints of the system are to try to minimize the total ranked preferences for all of the Capstone assignments. A student's ranking of a Capstone project is treated as the cost associated with a student/Capstone combination. There is currently a Relax IV-based algorithm called RelaxWin.exe that is used to solve this problem and arranges the students such that the total of the ranked preferences, i.e. the total of the ranks of the student's preferences, is minimized. The Pareto optimal solution for the assignments is found by minimizing the total cost of all of the students' rankings for the Capstone projects they are assigned to, while assigning a particular number of students to each Capstone. Here, we show how one can apply the RAPS design to the Capstone assignment problem. The resulting design is called the Capstone Resource Allocation Planning System (C-RAPS) interface.

Table 4. RAPS Information Visualization Techniques Applied to the Capstone Assignment Problem

System	RAPS	C-RAPS
Definition	Generic	Personnel assignments
Structure	---	Many to 1
Resources	Row	Students
Number	---	70-100
Details	Resource information box	Student information box
Name	---	Student name
Owner	---	Self
hard constraints	---	Each student to one project
soft constraints	---	Rankings
additional info	---	Special needs/desires
Example	---	Student Y should be assigned to Capstone X
Demands	Column	Capstone Projects
Number		20-30
Details	Demand information box	Capstone information box
Name	---	Capstone project name
Owner	---	Advisor
hard constraints	---	Number of students
soft constraints	---	How students ranked it
additional info	---	Special need/assignment
Example	---	Do not reassign students from this capstone project
Functions		
info retrieval	Hover Resource selection Demand selection	Hover Student selection Capstone selection
comparing characteristics	Color of cell Information boxes	Color of cell Information boxes
adding resource/ demand hard constraints	Alert notice	Alert notice
judge options	Color of cell Information boxes	Color of cell Information boxes
Make reassignments	Select + reassign	Select + reassign Drag and drop
multiple reassignments	Multiple reassignment window	Multiple reassignment window
Visual Information		
Find problems	Color of cell System overview	Color of cell System overview column
Resource/demand assignment	Text	Text
Hard Characteristics /Requirements	Filled cell	Filled cell
Soft Characteristics Costs	Color of cell	Gray –green – orange –dark red…
Characteristics met	Information boxes	Information boxes

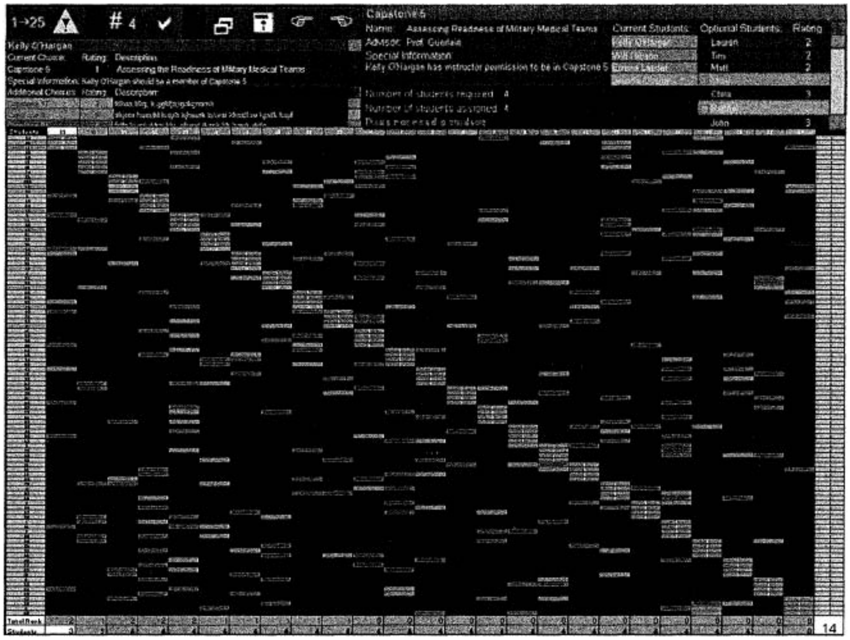


Fig. 18. C-RAPS Layout

The layout of the C-RAPS decision-support tool follows the basic RAPS design. The Capstone projects are considered the demands and are listed as the column names across the top, and the students are considered the resources and are listed as the row names down the left side (Figure 17). The figure displays 100 students, 25 Capstone projects, and the top four choices for each of the students. A number within a cell indicates that the student associated with the row of that cell is assigned to Capstone project associated with the column of that cell. The color of the cells indicates the ranking or cost associated with that cell's student/Capstone combination; the gray cells indicate a first choice, the green cells indicate a second choice, the orange cells indicate a third choice, and the dark red indicates a fourth choice. This color scheme was chosen to map the priority of desired assignments to the salience of the cells' color. The gray is the most desirable, and therefore the most salient. The dark red is the least desirable, and therefore the least salient. All of the cells that are black indicate that student/Capstone rankings for those projects are greater than the student's fourth choice. It is up to the decision-maker how many ranking preferences are presented at a time. In the figure, only the top four choices of the students are displayed, and an even less salient color than dark red would be used to represent any cell which is beyond the student's fourth choice.

5.2 C-RAPS Filtering and Sorting

The toolbar allows the user to increase the functionality of the C-RAPS system by controlling how the information is organized and filtered (Figure 18).

Fig. 19. C-RAPS Toolbar

5.3 Capstone Organization

The first two buttons labeled with 1-25 and an "alert" sign are the two organization methods for C-RAPS. When one of the buttons is highlighted in lighter gray that indicates that the user has the decision-support tool in that organizational manner. The 1-25 indicates a numeric organization of the Capstone projects. The "alert" sign indicates an attention method of organization where the Capstone projects that have the highest total ranked preference are listed first.

5.4 Filtering Options

There are two filtering options for the C-RAPS interface, indicated by the number (#) sign and the button with a checkmark. The button with the number sign is used to indicate how many ranked preferences are indicated on the screen. The number 4 next to the number sign indicates that the top four rankings for each student are displayed on the screen. This can be seen in Figure 18. To change the number of rankings displayed the user would simply click on the number sign button and choose the desired number from a drop-down menu. The checkmark button is used to filter out items that have already been assigned that the user does not want to change or view. By shift-selecting a group of Capstone projects and clicking the checkmark button, those selected projects and their assigned students would be minimized, leaving them visible to the user, but also increasing the space for the Capstone projects and students with which the user is concerned with (Figure 19).

5.5 Student Information Box

If the user wants more information about a particular student, or the Capstone rankings of a student, then the user can simply click on the student of interest and all of the relevant information pertaining to that student will be presented in the Student Information Box always visible on the screen (Figure 20). The Student Information Box includes information concerning the Capstone the student is currently assigned to, the ranking of the student's current assignment, a description of the currently assigned Capstone, and special information related to the student, and a list of the student's ranked choices and the Capstone's descriptions.

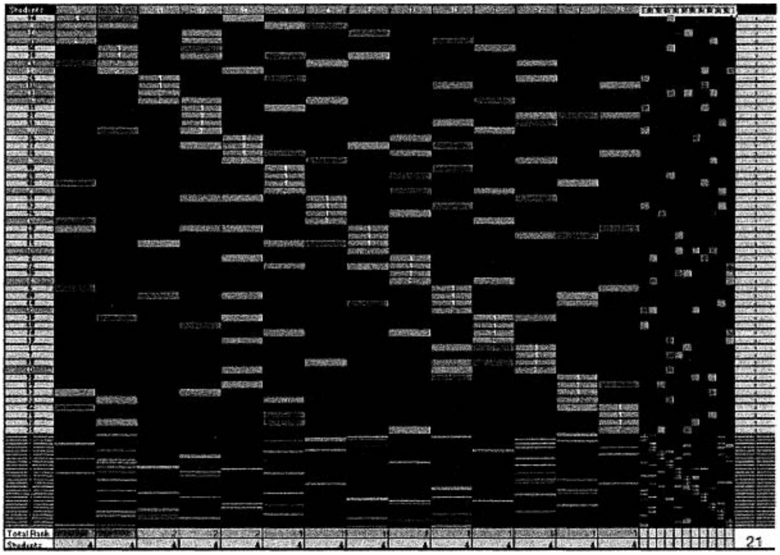


Fig. 20. Minimizing Selected Cells

5.6 Capstone Information Box

If the user wants additional information about a Capstone project, especially information concerning how students ranked the Capstone project then the user just needs to select the Capstone project he or she is interested in, and the Capstone Information Box will display all of the necessary information (Figure 21). The Capstone Information Box includes the name and description of the Capstone project, the advisor for the project, special information about the project, the number of students the Capstone requires, and the list of students currently assigned to the

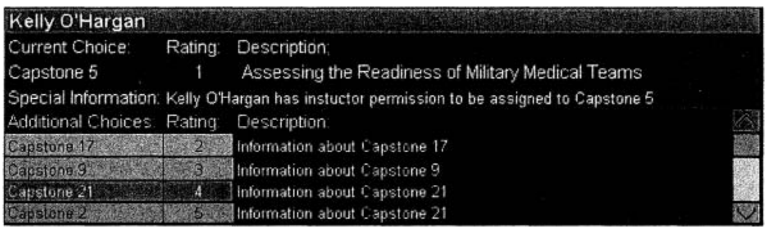
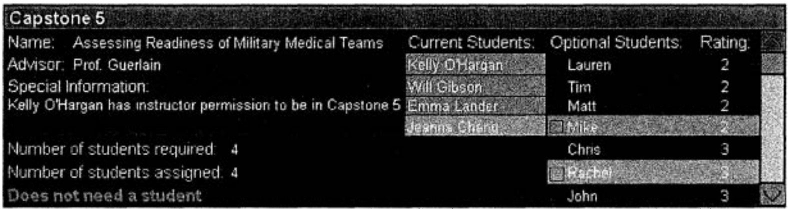


Fig. 21. Student Information Box

project. There is also a list of the optional students for the project, including the names of the students, and how they rated the Capstone project. Additionally, how each student ranked his or her current assigned Capstone is indicated by the color of his or her cell, gray for students currently assigned to their first choice, green for students assigned to their second choice, orange indicates third choice, dark red



indicates fourth, etc. In the example below Mike is currently assigned to his third choice project and Jeanna is assigned to her second choice project. If the user wanted Mike could replace Jeanna's spot by grabbing Mike's cell and dragging it over to Jeanna's position. If the user just wanted to add Mike to the list of current students the user would just drag Mike to the bottom of the list.

Fig. 22. Capstone Information Box

5.7 C-RAPS Optimization Tools

The last four buttons on the toolbar are used to help the user improve upon the Capstone assignment situation (Figure 18). When a Capstone assignment problem is created there are steps that the user goes through to find the optimal answer for the system. When the assignment tool in C-RAPS is first opened all of the students are assigned to their first choice projects, therefore some projects generally have too many students assigned to them while others do not have enough (Figure 22). Note that in Figure 36 the total ranked preference is 0, as shown in the bottom right-hand corner of the screen, but all of the needs of the Capstone projects are not being met. The total ranked preference (TRP) is the total cost, or in this case, the total number of ranks greater than 1. Since currently all of the students are assigned to their first choice the TRP is 0. For every student who is assigned to his or her second choice the TRP is increased by one, increased by two for third choices, and so on. The multiple reassignment window helps the user to de-conflict students that could fit the needs of different projects.

Next, the user should select all of the Capstone projects that need students assigned to them and click on the multiple reassignment tool. Students and their Current Capstone assignments are displayed in the first and second columns, (see Figure 23). The color coding in the second column helps the user decide which students are in oversubscribed projects and are thus good candidates for reassignment. After the user has selected how to best meet the needs of all of the projects and students, the user selects the reassignment button and all of the selected students will be reassigned to the associated projects. Note that at this point the total ranked preference is 14 with one Capstone and one student still in need of an assignment (Figure 24).

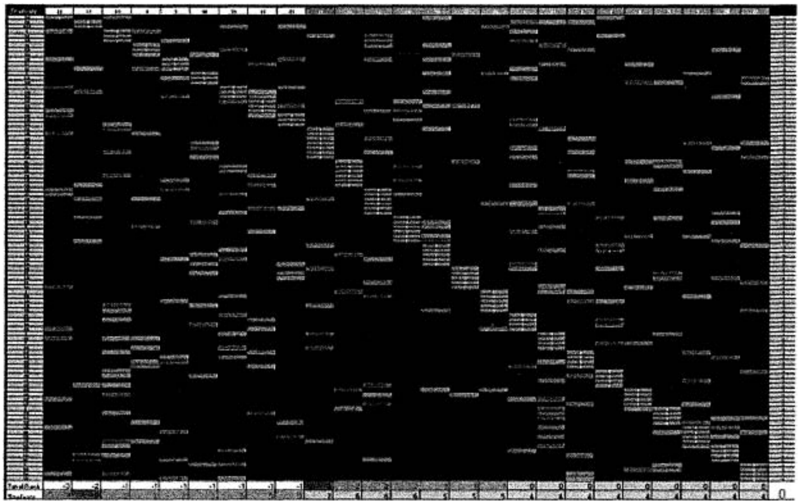
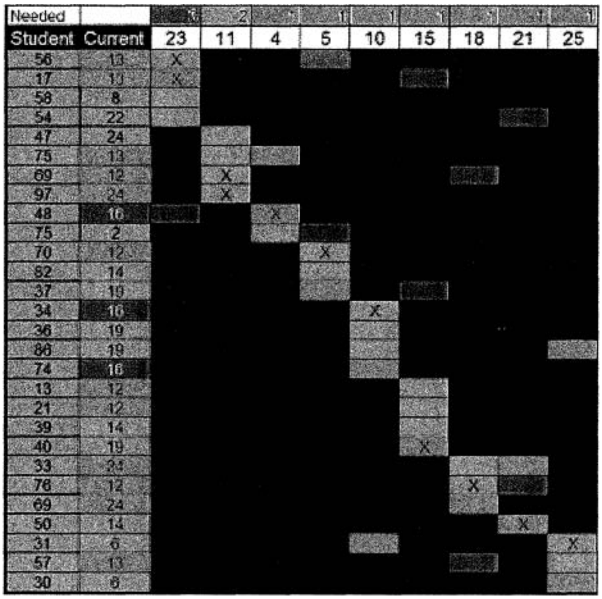


Fig. 23. Initial Capstone Assignments



Needed		[illegible]	2	1	1	1	1	1	1	1
Student	Current	23	11	4	5	10	15	18	21	25
56	13	X								
17	13	X								
58	8									
54	22									
47	24									
75	13									
69	12		X							
97	24		X							
48	16			X						
75	2									
70	12				X					
82	14									
37	19									
34	16					X				
36	19									
86	19									
74	16									
13	12									
21	12									
39	14									
40	19						X			
33	24									
76	12							X		
69	24									
50	14								X	
31	6									X
57	13									
30	6									

Fig. 24. Multiple Capstone Reassignment Window

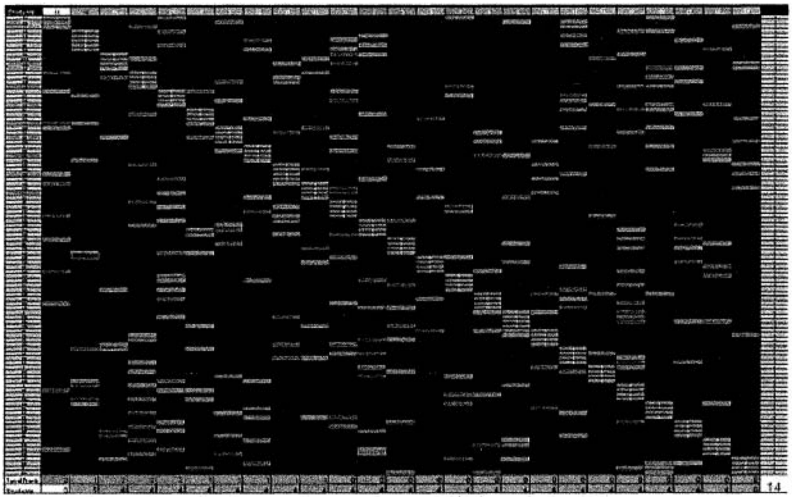


Fig. 25. Overview After Multiple Reassignment Window

The last three buttons on the toolbar are used to allow a user to "play" with the system and assignments and be able to go back to previous states of the system, (see Figure 18). At the point of the TRP being 14 the user should save the current state by clicking the button with a "floppy disk" on it. Then the user can click through and reassign students to different projects until all of the Capstone projects have the appropriate number of students assigned to them, and every student is assigned to a project (Figure 25). Note that the TRP is 21 and all of the demands of the system have been satisfied.

The user probably wants to save the system again at this point since a solution to the problem has been reached. Then if the user wishes to try to optimize the system further, he or she can continue to "play" with the system by clicking through the interface reassigning student A from Capstone X to Capstone Y, and then finding a different student B to take student A's position in Capstone X, and choosing a student C from Capstone Y to remove such that the appropriate number of students are included. In this example, by "playing" with the system for 15 minutes the first author was able to bring the total ranked preference from 21 to 18; further exploration revealed that there were multiple ways in which the students could be assigned to projects with a total ranked preference of 18, (see Figures 26 and 27). At this point the author saved for a last time and considered the system optimized.

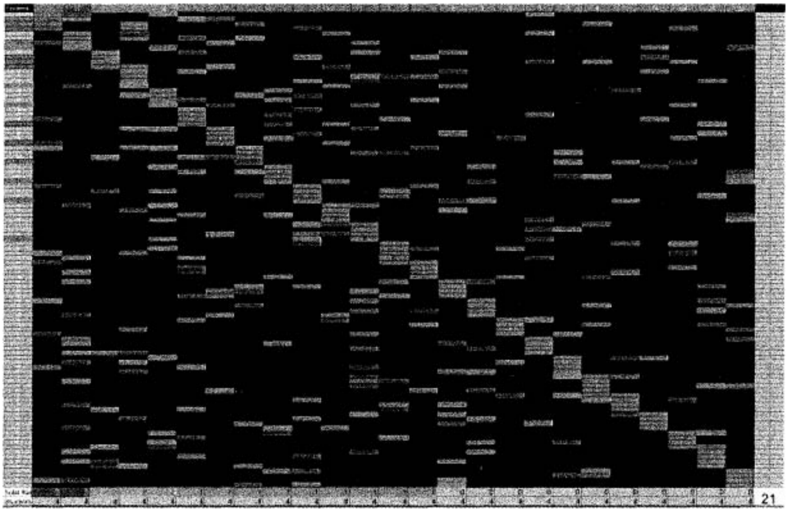
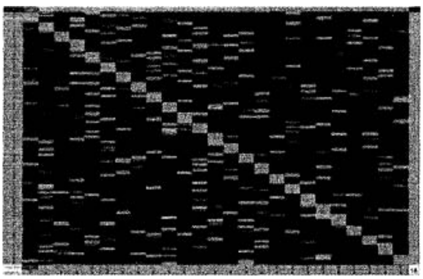


Fig. 26. C-RAPS With Complete Allocation



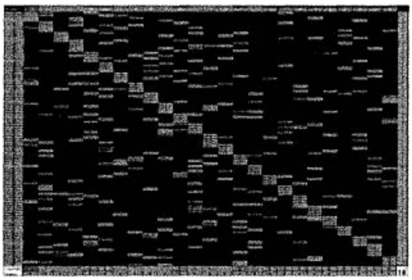


Fig. 27. C-RAPS Solution 18 (1)

Fig. 28. C-RAPS Solution 18 (2)

5.8 C-RAPS Accuracy

In order to establish how effective the C-RAPS interface is, the accuracy of the optimal solution was compared with the optimal solution found with the use of a mathematical optimization algorithm. The current method of assigning students to Capstone projects is for the data to be input into an algorithm and the solution that is output is the student assignments. If a user wanted to change the position of one student the entire optimization algorithm would have to be run again. The optimal solution the algorithm computed had a total ranked preference of 17, which was one TRP better than the C-RAPS solution of 18. Looking at the C-RAPS solution again, it would only take five reassignments to find the solution found by the algorithm, with a TRP of 17.

5.9 C-RAPS Utilization

Although it may seem as if a less optimal solution discounts the use of the C-RAPS interface, there are additional advantages to the C-RAPS interface that must be considered. First of all, there is a human decision-maker allowing that person to rate each change based on personal experience and the additional information included in the Student and Capstone Information Boxes. Secondly, the C-RAPS system is more flexible and easily adaptable to new information or preferences. For instance, if it turns out that three students need their assignments changed to put them in a particular Capstone project, the C-RAPS interface would allow the user to go back into the system manually and find the best way to adjust those three students without disrupting the rest of the assignments. However, if the same problem were to be solved by the algorithm it is unlikely that a similar output, disrupting as few students as possible, would be the result. Also, perhaps C-RAPS could be applied as an assistant to the current algorithm. For instance after a solution was found the C-RAPS interface could be used to go through the assignments to make judgment calls. If changes had to be made after the initial assignments, the C-RAPS interface could help with the decision-making process. Furthermore, there are generally a number of ways assignments can be made that have the same TRP optimal decision. C-RAPS could be used to find those other optimal decisions and then make a more informed decision about which solution to choose. These are just some of the reasons why the C-RAPS interface may be a useful tool for a decision-maker.

5.10 Size

The RAPS decision support tool allows the user to easily view more than 40 columns and 60 rows at a time, technically there is no limit as to how many resources and demands a user can be monitoring, since the system automatically adjusts to the size of the problem. With larger problems, some of the text can be hard for the user to read, but the hover function and select feature that expands the selected row helps the user navigate larger information spaces and the user can filter the data shown, and then use an attention-mode organization to help him or her more quickly determine the optimal resources to be reallocated. Additionally, the summary rows and columns allow the user to quickly assess the state of the problem solving, regardless of the size of the problem or what view the system is currently in.

5.11 Information Retrieval

There are a number of ways that the RAPS system makes information retrieval easier for the user. First, the system is designed to eliminate the need to manually search the decision-support tool by the incorporation of text search features, copy and drag-and-drop functions and hyperlinks. Also, since the interface may represent too much information to be able to be seen legibly, information retrieval is improved through the hover utility and the row/column expansion function associated with selecting a resource or demand.

6 Future Development

There are still many areas for further research and development of the RAPS interface. The basic RAPS design should be prototyped and improved through a heuristic evaluation and usability test of the complete system. An interactive prototype with real data and a back end simulation engine enables more realistic usability studies with domain practitioners as subjects trying to solve realistic scenarios.

Further work also needs to be done to determine how best to integrate the RAPS front end with algorithms that determine the options available and the rankings of those options automatically, such that the two systems could work together to make resource allocation assignments. Once one or more implemented versions of RAPS applied to specific domain problems exist, then such research could be conducted using the simulation as a test-bed for human-in-the-loop design experiments.

7 Conclusion

This paper provides the design community with a "generic" user interface design which can be adapted to several types of resource allocation domains. The features of the design include the ability to adapt to different size resource allocation problems, allow the user to vary the presentation of the information based on task demands, and to add in custom information about a problem. The design case study demonstrated how RAPS can be applied to the allocation of manpower (the capstone assignment problem). Further work remains to implement and test the designs in more realistic scenarios and to examine how/when resource allocation planning algorithms should be included in the system.

Acknowledgements

This research was funded by the Naval Surface Warfare Center, Dahlgren Division, under the guidance of Alan Thomas, through a grant from the Office of Naval Research Knowledge and Superiority Future Navy Capability Program

8 References

[1] Zhang, J. and Norman, D. "Representations in Distributed Cognitive Tasks". Cognitive Science. Vol.18. No. 1 Jan-Mar 1994.

[2] Woods, D. (1995). Towards a theoretical base for representation design in the computer medium: Ecological perception and aiding human cognition. In J. Flach, P. Hancock, J. Caird, & K. Vicente (Eds.), *Global Perspectives on the Ecology of Human-Machine Systems* (pp. 157-188). Hillsdale, NJ: Lawrence Erlbaum Associates.

[3] "Resource Allocation." Encarta Dictionary Online. World English Dictionary [North American Edition.] Microsoft Corporation. 2005.

[4] Laudon, K. C. and Laudon, J. P. Management Information Systems. 7th ed. New Jersey: Prentice-Hall Inc., 2002.

[5] Wickens, C., Lee, J., Liu, Y. and Becker, S. An Introduction to Human Factors Engineering,

[6] Guerlain, S. *Interactive advisory systems.* Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium, Savannah, GA, 166-171. 2000.

[7] Card, S. K., Mackinlay, J. D., and Shneiderman, B. Readings in Information Visualization Using Vision to Think. California: Morgan Kaufmann Publishers, Inc. 1999.

[8] Woods, D., Watts, J. *How not to have to navigate through too many displays* Handbook of Human-Computer Interaction, Second Edition, Elsevier Science, Amsterdam, 617-650, 1997.

[9] Rao, R. and Card, S. K. "A Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information." *Proceedings of CHI'94, ACM Conference on Human Factors in Computing Systems*, New York. 1994.
2nd Ed. New Jersey: Prentice-Hall, 2004.

Embedding complementarity in HCI methods and techniques – designing for the "cultural other"

Nielsen, Janni, Yssing, C., Levinsen, K., Clemmensen, T, Ørngreen, R. Nielsen, Lene

HCI research group, Department of Informatics,
Copenhagen Business School
Howitzvej 60, DK-2000, Denmark
janni.nielsen@cbs.dk

Abstract. Differences in cultural contexts constitute differences in cognition and research, which shows that different cultures may use different cognitive tools for perception and reasoning. The cultural embeddings are significant in relation to HCI, because the cultural context is also embedded in the methodological framework, the techniques and the tools that we apply. We lack a framework for discussing what and who we are, when we talk about a person as the user of an ICT system that has to be designed, developed and implemented. As a framework, we will suggest a theory of complementary positions that insists on solid accounts from all observer positions in relation to perspective, standpoint and focus. We need to develop complementary theories that embed complexity, and we need to reflect critically upon the forty years dominated by a rationalistic, empirical understanding of the user as illustrated in the literature and practice within the HCI paradigm in system development.

1 Introduction

The global digitalization of information and communication processes requires that the world citizens are literate in the use of computers. But the majority of the world populations are illiterates, and not only technical illiterates, but illiterates in the traditional sense of the word: they cannot read and write. However, the global ICT development largely disregards the problem with illiteracy and cultural differences.

India may serve as an example. India has developed an impressive ICT industry and has a very high level of expertise in software engineering. In addition, India has implemented e-government systems that also address the rural populations. But the Indian population is very large, and the potential users include highly diverse groups, many of which are illiterate. Experiments have shown that a gulf exists between the intended use of a technology and the actual use because "neither Development nor Quality Assurance Process consider Usability from the requirement phase or the pre-implementations phase" [1, 2]

Please use the following format when citing this chapter:

Nielsen, J., Yssing, C., Levinsen, K., Clemmensen, T., Orngreen, R., Nielsen, L., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 93–102.

One solution to the problems with illiterates explored by the Indian Government involves setting up electronic kiosks in remote areas and letting the electronic information process be handled by and through a kiosk operator - who may be a local administrator. India is divided into states, a state is divided into districts and districts are divided into blocks. A block may consist of 40-50 villages and a block administrator may be miles away – geographically and mentally – from the individual farmer in a remote village, who wants to ask experts in Delhi about the black spots on his crop. "In India, language, context, culture change in every few kilometres" [3]. The administrator may not know anything of the knowledge field in question, and the expert in Delhi may never have visited the remote area of the remote state in question. Villagers may have no concept nor understanding of computers and networks – and the technology makes no sense to them. The individual "user" becomes dependent upon the operator [3], and questions and answers may suffer from having to pass through the administrators. Besides, information is power, and the administrator's role as the gatekeeper of technology, interpreter and handler of information may undermine the intended technological enhancement of democracy, as gate keeping may develop into a very powerful (and misused) position.

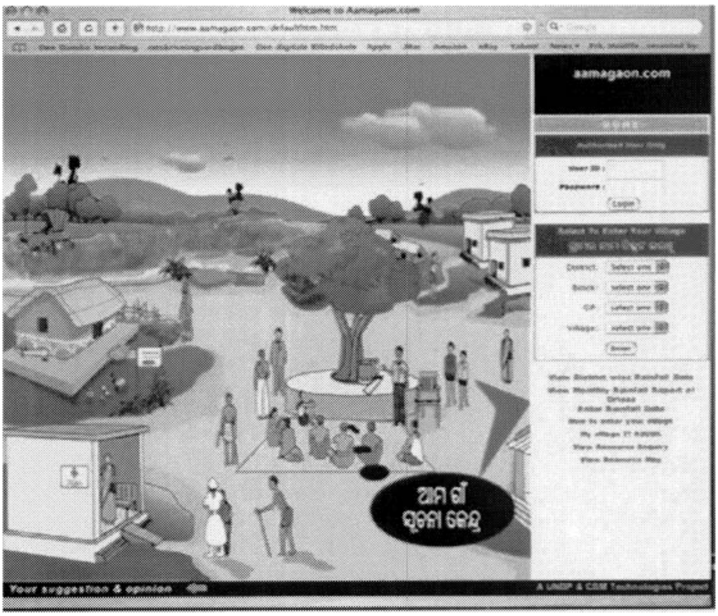


Figure 1. Illustration: example of an e-government web site: Rural Planner

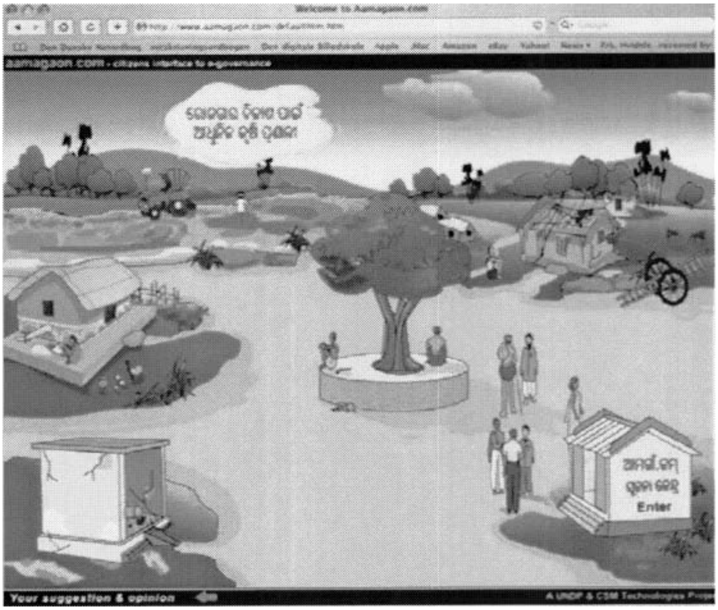


Figure 2. This web page may be activated through mouse over. This means that a text bubble will occur when the user/operator moves the cursor over an object on the screen. In this case, he has moved it over the tractor in the upper left hand corner and a text has popped up.

If the user/operator moves the cursor over the tree, several items become visible: people, the health station, a text in a black bubble and a red arrow that points to a menu bar on the right side. The user/operator is asked to key his user id, password and entry (location), where he can choose between district, block, group or village, and then he may select to see rainfall for given periods.

There is a digital divide between those who have access to IT and those who do not, those who can read and those who cannot, those who speak English and those who do not [4]. Different solutions have been suggested and prototypes developed, e.g. "interactive speech interfaces" [5] and special navigational assistance such as "signboard system, vocal agents or natural language processing dialogue" [6].

Another solution has been to suggest personalized e-government services, and experiments have been carried out with "personalized services through touch screen kiosks" to the illiterate villagers. But there are problems with "establishing identity of person and verification" [7]. In one experiment, potential illiterate users were asked to choose a combination of images, 7 images for their username and another 7 images for their user identity. There was no problem in getting the users to choose among the many different visual images, which differed greatly in style and size. However, a few days later the users did not remember all the visual images, which they had chosen, or the sequence in which they were chosen. In another experiment, villagers who were unfamiliar with computers were unable to use the keyboard despite careful instructions. The researchers concluded that the users' perceptual-motor skills were not developed to handle small keys on a board. It raises the question whether one can touch and interact with something in a meaningful way if the object and the actions do not make sense?

1.1 A cultural bias

A main problem seems to be the relation between the culture of Information and Communication Technologies and the cognition of everyday life. The villagers had no problems reflecting on rain, clouds, grey skies, sun, etc. in concrete experiences from everyday life. But when these objects were transformed and visualised on a computer screen, they did not recognize them and were unable to talk about them when interviewed. They were visualised, but still abstract - not concrete experiences like seeing the black spots on the crop. " We do not exactly know the information need and information seeking behaviour of the rural populace" [2], and we do not know their reasoning on or perception of the ICT applications, to which they are introduced. This may be difficult to understand for academics, because abstract concepts and meta-reasoning are so fundamental in our professional lives. But reasoning and thinking based on the concrete experiences from everyday life cannot capture the meta-reflections embedded in the world of ICT applications.

Context is embedded in cultures, and differences in cultural contexts imply differences in cognition [8]. This understanding has to be taken one step further as research shows that cultures may use different cognitive tools for perception and reasoning and there are culture specific differences in the way that people think and reason [9]. A logically true statement may be true in English, but not in Hindi, or Chinese.

The cultural embeddings are significant in relation to HCI, because the cultural context is also embedded in the methodological framework, the techniques and the tools we apply. The HCI field fails to consider the role of culture in its methods and techniques [10], but they cannot escape a cultural bias. Traditional HCI methods and techniques have developed along with the IT industry and are based on western thinking.

2 Representation of Users

In computer applications, designers have long used representation of users. A recent example of the representation of humans can be found in Microsoft OneNote ®[1] software, where users are represented by portraits (photos) in usage scenarios know as personas [11, 12, 13]: On OneNote's Danish website, Kirsten is a consultant, Søren is an engineer and Kathrine is a student, who takes notes in English although she is a Danish student. On the German site, she is named differently, but the photo and task are the same. The diversity of people's skin colour in the different usage

[1]Microsoft.com, (retrieved Jan. 21 2004), http://www.microsoft.com/office/onenote/prodinfo/usage/journalist.mspx

scenarios shows that the company addresses "equity issues", but it applies usage scenarios with an embedded representation of users as mono-cultural and function-oriented ideal types. Thus, we are all on a global scale exposed to descriptions of a limited number of ideal humans who apply technologies in certain ways and are blind to cultural differences and illiteracy.

Not even the representation of the user in the traditional Human-Computer Interaction (HCI) techniques and methods reflects a complex and differentiated understanding of human beings. In most of the Human Factors' representations [14, 15, 16], it is not a person who is represented, but computer applications with a one-dimensional user as an appendix [17, 18, 19]. Despite conscious and explicit attempts to get around the one-dimensional human being, even the new interaction design research [20] ends up with a simplified, rational subject, and interaction remains something that takes place in a closed space: within the human head. When Human Factors as well as Interaction design focus on tools, techniques and methods, they do not have a clear understanding of the underlying theories, and hence they cannot frame the use of tools in the embedded world views.

3 A challenge to HCI

The challenge lies in developing more diverse representations of the complex human being in an information and communication technological (ICT) perspective. Inadequate descriptions of humans are decisive for the designer's conception of the user and will eventually govern the development of the user interface [21]. Hence, they also have an impact on the user functions designed as part of the systems, and they influence the human-computer interaction - and the human beings that use the systems [22]. As such, the designers' user representations influence our conceptions of what humans are and what computers are, and in the end, they will also influence our imaginations about the future society as a whole [23, 24]. Besides, the inadequate descriptions of users do not enable or support the design of a future ICT that is oriented towards humans as individual users in other cultures and contexts than the standardised work and mass consumption culture.

In our opinion, we lack a richer and more complex description of who we design for and what they will do with our designs. We lack complementary methods and techniques to develop complex descriptions of the future systems' users, and we lack methods and techniques to develop complex user centred designs, tests and evaluations [25]. Our claim is radical: We need to develop complementary theories that embed complexity, and we need to reflect critically upon the forty years of dominance by a rationalistic empirical understanding of the user expressed in most of the literature and practice within the HCI paradigm in system development.

4 A floating context and perceptual interaction

Users do not identify with – and cannot be identified from a traditional demographic categorizing of sex, age, profession, etc. We are immersed in different cultures and take on roles and functions depending on which contexts we enter into and are cocreators of. This also applies to cooperation and communication technologies. We may play with our identity in chat rooms; we can cooperate with colleagues via the net and then, a few minutes later, log in and be a student in a virtual master study programme. However, within ICT the representation of the human has been based on a rational ideal that is goal oriented, information seeking and task directed [26, 27]. Quantitative segmentations have played a major role, and because computers were developed for standardised work (e.g. text editing) and mass consumption, the human had to become someone who could adapt to each new generation of software, instead of the other way around. It still characterizes computer use (except for front users) that humans have to adapt. At the same time, however, ICT is spreading into people's everyday life and all other aspects of life, both in specific, personal ways and as general, cross-personal globalization. As a consequence, technologies will have to work in ambient contexts defined by the different ways and areas and the different uses. The context becomes *floating*: I am physically present at my office, in my chair, and, at the same time, I am present on the net, virtually present in Bangalore, walking down the 'MG road' deep into discussion with an Indian colleague, sensing the noise from the traffic, the chaotic street, the multicoloured flower-arrangements in the many small shops – and aware of the two students who enter my office and place a book they want to return on my desk.

The children of today will be the power-users of tomorrow. They are emotionally engaged and develop new cognitive skills [28]. Without efforts, they navigate deeply into the application transferring to other applications, and all the time they have an overview and know their way "home". In this development, we find a challenge for research. The interaction with the computer is mental. The computer interacts directly with the human cognitive processes: perceptual, emotional, sensual and conceptual. Hence also the sensual, visual and emotional interaction, which relies on tacit processes [29] and takes place above, around and below the verbal and written interaction, becomes significant. But how do we create and communicate this knowledge about humans' use of technology? How do we use these creations to design software and interactive products? It is not only the goal directed interaction we need to understand, design and evaluate, interaction also embeds aesthetics and pleasure [30]. Irrespective of the technological goals, the intentions with "pervasive, ubiquitous and transparent computing" [31, 32] are identical: Technologies should be "unobtrusive", i.e. we should not focus on the technology, but on the activity we are currently doing.

We suggest that Human-Computer Interaction (HCI) research should contribute to the design of future ICT systems by focussing on (1) culture and floating contexts (2) the double complexity of complex roles and functions and (3) the cognitive basis of the interaction with the computer.

The research challenge lies in conceptualizing and representing the complexity. It should be represented in the methods and techniques for analysis, design, test and evaluation of human-computer interaction. The conceptualization of research objects is all framed by culture through its embeddedness in our understanding of humans, theory and technology. To analyse this complexity, we need to apply a theory of complementary approach.

5 A complementary methodology

The cultural frame, the complex human being, the floating contexts and the mental interaction cannot be described from one single observer position. They may eventually be described and presented in a richer diversity by combining many observations from many observer positions. What we need is a framework for discussing what and who we are, when we talk about a human as a concrete user of a concrete ICT system that has to be designed, developed and implemented. As a framework, we suggest a theory of complementary positions, which insists on solid accounts and theoretical explanations from all observer positions in relation to perspective, standpoint and focus. The framework enables us to relate to the observers' influence on the observed aspects [33] and the limitations encountered by culture and language(s), when the subject-object distinction cannot be maintained.

Adopting a theory of complementary positions as a framework necessitates an experimental approach. This allows the representations of the Human in HCI design methods and techniques to be tested and developed in iterations during the whole development and use process. As a point of departure, we have developed the figure below. The model shows examples of areas, within which different types of human representations are needed. They have to be further investigated, both on the level that concerns one technique within one phase of the system development, but also on the level of methodological approaches to ICT across the whole development process and user cycle.

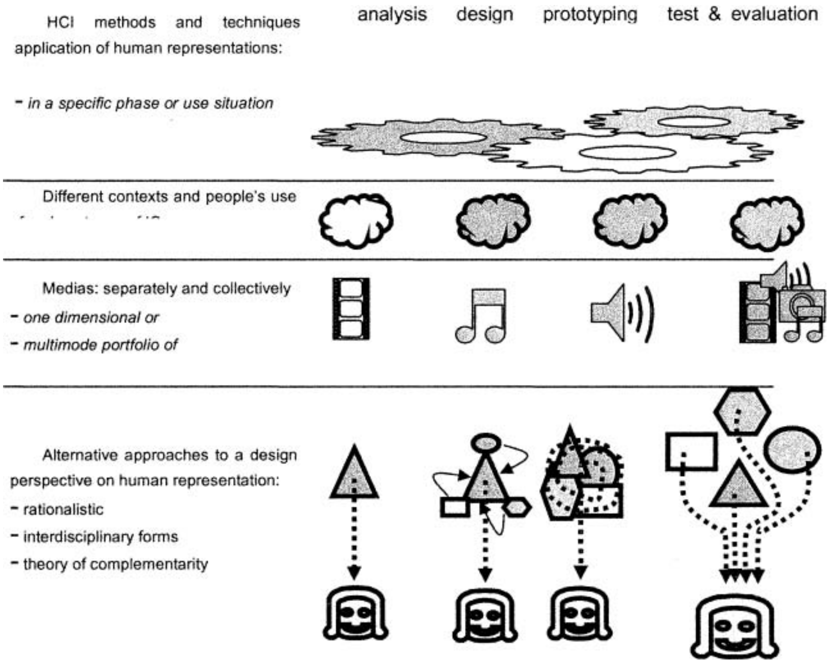


Figure 3. Context sensitive HCI design methods and techniques based on an experimental approach.

6. Future work

As the complexity of the Indian scenario has shown, highly creative approaches to development from a user perspective are necessary. We have to reveal cultural biases embedded in IT applications and must have an open mind for development of HCI methods and techniques as well as new applications. But the design and development need to be based on experimental sketches and prototyping, just as techniques and tools for test and evaluation of human interaction with the computer/other ICT artefact have to be developed on an experimental basis. Confronting existing techniques and tools, e.g. contextual enquiry, cultural probes, scenario development, the technique of engaging persona, iterative prototyping, design of icons and graphical (dynamic) interfaces to applications with explorative and experimental approaches, may lead to innovative designs.

Our design approach attempts to integrate a focus on analysis and design that goes beyond the general reliance on iteration as a way to develop products that fit the user's needs and context. In a period with flexible, mobile technologies used in drifting contexts, it is vitally important to maintain a focus on users and the complex user situations. As society and users' work become increasingly complex and global, we believe complementary techniques resulting in multidimensional user descriptions may lead to a focus on a robust and diverse user approach, for example by means of extensive work studies providing multidimensional rich portraits of users.

References

[1] Jani Rahesh and Vaishali Badave (2004) Usability in Quality Assurance Process, in *Proceeding of The First All India Human Computer Interaction Conference*, Bangalore, India 6-7 December 2004

[2] Singh Jitender and Gajendra Agrawal (2004) Information Need and Dissemination: Indian Rural Context in *Proceeding of The First All India Human Computer Interaction Conference*, Bangalore, India 6-7 December 2004

[3] Parmar, Vikram Singh and Prasad Wani (2004) Hardware browser "Net Babu" – An approach to internet browsing for novices in *Proceeding of The First All India Human Computer Interaction Conference*, Bangalore, India 6-7 December 2004

[4] Yajnik, Nilay M., Digital Divide and Human Computer Interface, Workshop, *Proceeding of The First All India Human Computer Interaction Conference*, Bangalore, India 6-7 December 2004

[5] Girija, P.N. (2004), Design of Interactive Speech Interfaces,, *Proceeding of The First All India Human Computer Interaction Conference*, Bangalore, India 6-7 December 2004

[6] Panwar, Vikas & Yammiyavar, Pradeep, Navigational Assistance and its Usability in Virtual Environments, Workshop, *Proceeding of the First All India Human Computer Interaction Conference*, Bangalore, India 6-7 December 2004

[7] Kaatre Dinesh S.(2004) Using Mnemonic Techniques as part of Pictorial interface for Self-identification of Illiterate Villagers, in *Proceeding of The First All India Human Computer Interaction Conference*, Bangalore, India 6-7 December 2004

[8] Berry, JW & Dasen PR (1974). *Culture and Cognition: Readings in Cross-Cultural Psychology*, London: Methuen

[9] Nisbett, Richard E. (2003) *The Geography of Thought – How Asians And Westerners Think Differently – And Why*, Nicholas Brealey Publishing, London

[10] Smith A. ad Yetim, F. (2004) Global human-computer systems: Cultural determinants of Usability, Editorial, *Interacting with Computers, vol. 16*

[11] Mikkelson, N., & Lee, W. O. (2000, August 14 - 18, 2000). Incorporating user archetypes into scenarios-based design. Paper presented at the *UPA 2000* - Designing the New Millenium Today, Asheville, North Carolina.

[12] Nielsen, L. (2002). From user to character. Paper presented at the *Designing Interactive Systems*, London.

[13] Pruitt, J., & Grudin, J. (2003). *Personas:Practice and Theory* [pdf file]. Retrieved March 4, 2003, from the World Wide Web: http://research.microsoft.com/users/jgrudin/

[14] Baecker, R., Grudin, J., Buxton, W., & Greeenberg, A. (1995). *Readings in Human-Computer Interaction: Toward the year 2000*. California: Morgan Kaufmann Publishers.

[15] Dix, A., Finlay, J., Abowd, G., & Beale, R. (2004). *Human-Computer Interaction - third edition*: Prentice Hall.

[16] Lindegaard, G. (1995). *Usability testing and system evaluation*. London.

[18] Card, S. K., Moran, T. P., & Newell, A. (1980). The Keystroke-Level Model for User Performance Time with Interactive Systems. *Communications of the ACM, 23(7), 396-410.*
[18] Nielsen, Jacob (1992). Evaluating the thinking aloud technique for use by computer scientists., *Advances in human-computer interactions (Vol. 3).* Norwood: Ablex Publishing Corp.
[19] Nielsen, J., Clemmensen, T., & Yssing, C. (2002). Getting access to what goes on in people's heads? - Reflections on the think-aloud technique. *Proceedings of the NordiCHI 2002, Århus, Denmark.*
[20] Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction Design: Beyond Human-Computer Interaction:* John Wiley & Sons.
[21] Kumar, K., & Bjørn Andersen, N. (1990) A Cross-Cultural-Comparison of IS Designer Values, *Communications of the ACM, 33(5), 528-538.*
[22] Levinsen, K.T. (2003) System development influenced by Rituals and Taboos *.Proceeding of HCI International Crete.*
[23] Weizenbaum, J. (1976) Computer power and human reason. San Francisco: Freeman.
[24] Winograd, T., & Flores, F. (1988) *Understanding Computer and Cognition - A new foundation for Design.* N.Y.: Addision-Wesley.
[25] Karat, J., & Karat, C. M. (2003). The evolution of user-centered focus in the human-computer interaction field. *IBM Systems Journal, 42(4), 532-541.*
[26] Levinsen, K. & Orngreen, R. (2003): Locating Students' Competencies - A Prerequisite for Collaboration, *Proceedings from 2nd European Conference on e-Learning (ECEL 2003),* November 2003, Glasgow, p. 261-266.
[27] Ericsson, K. A., & Simon, H. A. (1984). *Protocol Analysis. Verbal reports as data*: Cambridge Massachusetts.
[28] Nielsen, J. (2003). Visual cognition and multimedia artifacts. In O. Danielsen & J. Nielsen & B. Holm-Sørensen (Eds.), *Learning and narrativity in digital media*: Samfundslitteratur.
[29] Nielsen, J., Christiansen, N., Clemmensen, T., & Yssing, C. (2003). Mindtape - a Technique in Verbal Protocol Analysis. *Procedings of HCI International 2003, Crete.*
[30] Jordan, P. W. (2000). *Designing pleasurable products.* London: Taylor & Francis.
[31] Weiser, m. (1998) The Future of Ubiquitous Computing on Campus. *Communications of the ACM, 41, 41-42.*
[32] Weiser, M., & Gold, J. S. B. (1999) The Origin of Ubiquitous Computing research at PARC in the late 1980s. *IBM Systems Journal, 38(4), 693-696.*
[33] Allen, G. (1959). *Heisenberg, W. - Physics and Philosophy*: Unwin Edition.

Using Sketching to Aid the Collaborative Design of Information Visualisation Software - A Case Study

Brock Craft, Paul Cairns

UCL Interaction Centre (UCLIC), Remax House, 31/32 Alfred Place
London WC1E 7DP, United Kingdom
brock@craft.org
http://www.uclic.ucl.ac.uk

Abstract. We present results of a case study involving the design of Information Visualisation software to support work in the field of computational biology. The software supports research among scientists with very different technical backgrounds. In the study, the design process was enhanced through the use of sketching and design patterns. The results were that the use of sketching as an integral part of a collaborative design process aided creativity, communication, and collaboration. These findings show promise for use of sketching to augment other design methodologies for Information Visualisation.

1 Introduction

The field of Information Visualisation (IV) is concerned with the development of software that allows people to visually explore and understand complex datasets. These tools use various techniques for the presentation of information that range from those used in graphic communication [1] to complex interactive systems that represent data in multiple views, possibly even whilst that data is being fed into the system [2]. At the heart of these interactive systems is the principle that active engagement with the on-screen representations takes advantage of the visual processing capacity of the brain to amplify cognition [3]. IV systems may be either standalone or web-based applications. Currently, some online examples of these IV systems are available at: www.hivegroup.com, www.smartmoney.com, and www.cs.umd.edu/ hcil/research/visualization.shtml.

As with any complex software, the creation of Information Visualisation tools is difficult and depends on the general software concerns of fitness for purpose, context of use, modes of interaction, integration with existing workflows as well as the specific concerns of how and what to represent visually, and how to make the interaction available and intuitive to the user. In the field of IV, there are few methodologies that prescribe a suitable approach to the development of IV tools. Instead, developers appear to rely primarily upon guidelines or on inspiration from the many novel and effective visualisation tools that have already been developed [3]. This is particularly true for those who are not familiar with the techniques and principles of IV.

Please use the following format when citing this chapter:

Craft, B., Cairns, P., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 103–122.

A surprising gap in these guidelines and descriptions of development is the absence of detailed discussion concerning the role of sketching and drawing when designing new IV tools. As visual representation is so central to the nature of IV, we hypothesize that sketching almost certainly occurs, however its role, relevance, and purpose in the design process is not discussed. This suggests that, when used, it is done so in a free-form manner rather than in some more disciplined and methodical manner as a key part of the development process. By contrast, in more mature design processes such as graphic communication and architecture, sketching is used as an integral and systematic part of the design process [4, 5]. And while sketching has been discussed in generic user interface design – paper prototyping can be seen not only as a fast prototyping technique, but also as 4-d sketching [6] – the field of IV has largely ignored its benefits. The overall aim of this work, then, is to consider the use of sketching in the development of Information Visualisations with a view toward suggesting how it can be used effectively and systematically to support developers.

This paper reports on a particular case study in which an IV tool was developed to support the communication between biologists and mathematicians in a computational biology research project. Extracting valid research conclusions from such a case study is always problematic. There is necessarily, as with any complex software development process, no comparable development process in terms of timing, goals, team, and organisational context. Nor is it appropriate that the project be allowed to fail because of intrusive adherence to the research protocol. The approach we have taken therefore is to use Action Research (AR) [7] to aid and support the developers whilst deriving knowledge in the form of a case study. Hopefully with this approach, as will be discussed below, we are able to extract insights into the use of sketching in designing IV tools that not only supported the team we were working with but also have significance beyond this study.

Given the general craft-approach to designing IV tools, we used a framework of IV patterns proposed by Wilkins [8] as a way to introduce IV concerns to the case study team. These patterns are intended to function as design patterns in the architectural [9] and software engineering sense [10, 11] and thereby lead to effective IV development. Aimed at improving usability, they are based on the guidelines and examples that are commonly referred to in IV research and hence, presented a suitable starting framework for the team. It was felt that this approach would bring a structure that would support and encourage sketching around key IV ideas.

The key findings from this case study were that sketching substantially aided the development team in creating their visualisation software. Specifically, sketching helped team members in three key areas: Communication, Creation, and Collaboration. Sketching improved Communication by allowing team members to simultaneously share ideas verbally and visually, to clear up misunderstandings, to build up simple ideas into complex ones, and to record the activity for later reference. Sketching improved Creation by helping people to think about design problems in new ways, and to work out novel solutions to problems. It also helped them to overcome the intimidation of the design requirements and to overcome "mental blocks". Finally, sketching supported Collaboration by helping team members to share complex ideas with the group in a common space, by facilitating group input on new ideas at the time of ideation, and by helping the designers to suggest and try out interaction ideas with the group through scenarios. With the aid of a body of knowledge about effective IV solutions in the form of IV design patterns, the group were able to determine concrete solutions to the design problems that they had identified in the requirements gathering phase of the project. The result was that whereas before the sketching intervention the participants had only limited ideas about how to proceed with their project and were not able to identify a way forward, after the intervention the team had proposed eight specific Information Visualisation software modules as strong candidates for development and had prioritised exactly which of those projects would yield the most benefit to the end users and project stakeholders.

With regard to more general IV software development, the results of this work seem to indicate that it is fruitful to systematically employ sketching in the IV design process. Effective use of sketching as applied in this case study can lead to creative, visual solutions. This is especially relevant to Information Visualisation, as IV relies upon novel visual representations to enhance cognition of information. Particularly in the early phases of ideation and brainstorming, sketching helped team members turn a set of requirements into concrete ideas for visual representations and interactions. This helped them to overcome obstacles to creativity and to effectively address the needs of the project during the design phase.

2 Research

The case study involved the creation of software to support the working practice of biologists and mathematicians on a Computational Biology research project at University College London. Known as the Beacon Project, its goals are to create computational models of biological processes in the human liver, such as metabolisation of glucose. It is hoped that such models will help biologists to better understand liver function and to perform experiments that are not possible in the lab.

The need for Information Visualisation software arose out of problems that Beacon Project members encountered in the course of their work. Most acute among these was that the project team members could not effectively collaborate on developing models because this requires very specialized knowledge in rather different fields of research. Computational models that can begin to describe liver function are of a level of complexity that requires the knowledge and skills of specialist mathematicians. Similarly, intimate knowledge and intuitive understanding of liver metabolisation require many years of biomedical training and research. As the Beacon project matured, a challenge became apparent. The project mathematicians needed to know that the models they were in the process of developing were accurately expressing liver function. They also needed the biologists' help in determining where to make improvements when those models were inaccurate or wrong. But they could only obtain this guidance with the collaboration of biologists who were not able to understand the mathematics involved at a sufficient level of specificity to make useful recommendations. The group decided that a software solution might address this problem, if it would help in bridging this knowledge gap.

The Beacon team determined that the best way to do this would be to express the activity of the models visually and interactively which would aid understanding and cognition of all the team members; in short, they needed Information Visualisation software. However, computer scientists on the team did not have any experience in designing such tools. The Beacon project team contacted researchers in Information Visualisation at UCL Interaction Centre (UCLIC) to address these challenges and to aid them in arriving at a design for possible solution. The UCLIC team offered to aid the Beacon team in the IV design process and with their agreement, to use it as an opportunity for a case study in the design of Information Visualisations.

2.1 Method

As the research involved the study of human design processes in an information systems work environment, we used a qualitative and collaborative research method called Action Research (AR) [5]. Based upon post-positivist and interpretive research paradigms, the goal of Action Research is to derive useful empirical knowledge through the study of human phenomena in the environment in which they occur. Moreover, rather than isolating the researcher as a disinterested observer, AR situates the researcher squarely in the environment under study and invites the collaboration of both researchers and subjects as co-participants in the enquiry. The philosophy of the AR approach is that some knowledge about human activity can best be gained from the natural environment in which phenomena occur and that the acknowledged participation of the researcher and the subjects as co-participants are necessary for understanding them. Furthermore, AR recognizes that the researcher has knowledge that may be relevant to the activities under study and may contribute this to the research setting. Indeed, AR is particularly suited to studying processes where the organisation can be aided by the expert knowledge of the researcher. Thus, the visualisation design process in the Beacon Project which comprises the case study research reported here was done in a participatory and open manner. We believe this was particularly well-suited for understanding how sketching is useful early in the visualisation design process. Another method, such as conducting research in a controlled laboratory setting would not have yielded the kind of knowledge derived from real-world design problems.

In terms of the software development methodology, we used an approach based upon generic, Human-Computer Interaction oriented frameworks of iterative software design [12], which we call the Information Visualisation Design Framework (IVDF). Figure 1 shows the IVDF development process. It involves Requirements Gathering, Visualisation Design Activities using sketching and design patterns, Prototyping, and Evaluating. It situates sketching and the use of design patterns early in the development process and importantly, encourages this activity at the stage where ideation and creativity are involved. The focus of this case study was on the use of sketching and design patterns in the Visualisation Design Activities phase.

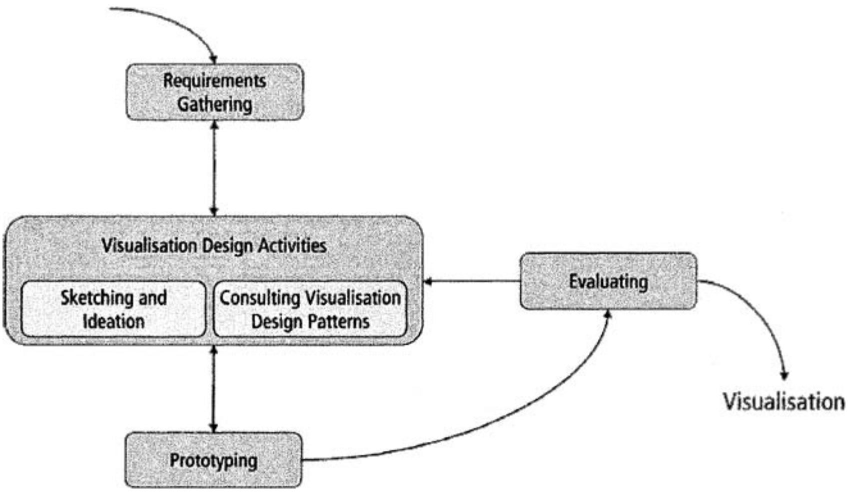


Fig. 1. The Information Visualisation Design Framework.

The first part of the IVDF is the Requirements Gathering process. Like requirements gathering techniques from other HCI-oriented development approaches [13], this involves collection of data from all stakeholders, whether by observation, surveys, interviews, or other means. We used personal interviews with project team members and end-users as the primary method for gathering requirements. Combined with design specifications that had previously been identified by project managers, these formed the Requirements that were used to guide the design sessions. Prior to the Design Activities Phase, these two sources were combined and it was agreed with the project team that these would be the Requirements used for development.

The Design Phase sessions were then held in several instances over a three month period. Meeting in small groups of not more that six people, willing participants from all parts of the Beacon Project engaged in the activity with the facilitation of a researcher from UCLIC. The design sessions were held at the Beacon Project premises in a conference room that was relatively free from distraction. They were typically two to three hours in duration including breaks. Sketching materials, such a large pad of heavy A3 drawing paper and various coloured writing tools were provided by the facilitator. It was agreed that audio from the design sessions would also be recorded and transcribed for later analysis.

The UCLIC researcher mediated the design process and invited active participation from the team members. In keeping with the AR methodology, the UCLIC researcher also offered knowledge and experience about effective IV solutions, where appropriate. Design patterns were used to demonstrate useful IV techniques and to aid brainstorming. Also, where appropriate the facilitator actively encouraged participants to use sketching tools during the design process and indeed, used these tools to elucidate ideas and explore possibilities with the group. To aid the research, active feedback about the design process was elicited where appropriate. In this way, the design sessions were completed by collaboration of the researcher and the participants and with the acknowledgement that the design activity was part of a research process.

2.2 Results

After completion of the Design Sessions, the data, in the form of the sketches and the audio were analysed. To aid analysis, some of these data were presented to the team members in follow-up interviews. In many instances, verbatim transcripts of the design sessions were used to gain feedback from participants about the nature of the design process and about how sketching, the design patterns, and the activity of the UCLIC facilitator as subject matter expert, were useful. Further analysis revealed patterns of activity that the sketching made possible.

Drawing from the audio recordings, design sketches, and follow-up discussions, we have identified three categories of activity that sketching facilitated: Communication, Creation and Collaboration. These categories emerged from the many different ways that team members used sketching in the design process. We have chosen highlights from the case study to illustrate how sketching helped Creation, Communication, and Collaboration. These highlights are presented below.

For each of the examples, we have excerpted some of the dialogue recorded during the design sessions and have provided the corresponding sketches that participants created. For the sake of brevity, only the first letter of participants' names is used in quoted conversations. While the data from the design sessions contained many more examples than could be presented in this paper, we have chosen highlights which we believe to be most representative.

2.3.1 Communication

Perhaps the most significant benefit that sketching brought to the design process in this case study was enhanced communication. The adage "a picture is worth a thousand words" is a good summary of the way that sketching helped in the communication of team members. One of the most difficult parts of designing in groups is ensuring that everyone understands the complex issues under discussion. Sketching helped to communicate new ideas quickly, to support verbal communication, to confirm understanding, to form a written record for later reference, to build complex ideas, and to explain difficult concepts. The following examples from this case study show how Communication was supported by sketching and how it facilitated the design process.

Communicating ideas verbally and visually, triggering understanding. In this example, a participant uses sketching to explain how the interface will graph particular mathematical variables in the computational model. This helps a mathematician on the team, J, to understand how the software will implement constraints on variable output and make those apparent to end users in an interactive graph. Note that to emphasise his point, P adds the words "Min" and "Max" to the sketch during his verbal explanation. Refer to Figure 2.

"P: This is basically what in your interface shows you a graph of a particular variable – I'll draw it again, I suppose. So this is your model view, and you have your variable tab. And we know there are problems with tabs, but. And we have the name of the variable here, which is *P* subscript *r*. And then it's got a graph here. And what we want to be able to do is add a button here that gives us some sort of interpretive result for that. So what we want to do is do "Max" and "Min" and then it would go "Max", "Min"."

J: Oh, that was important, yeah.

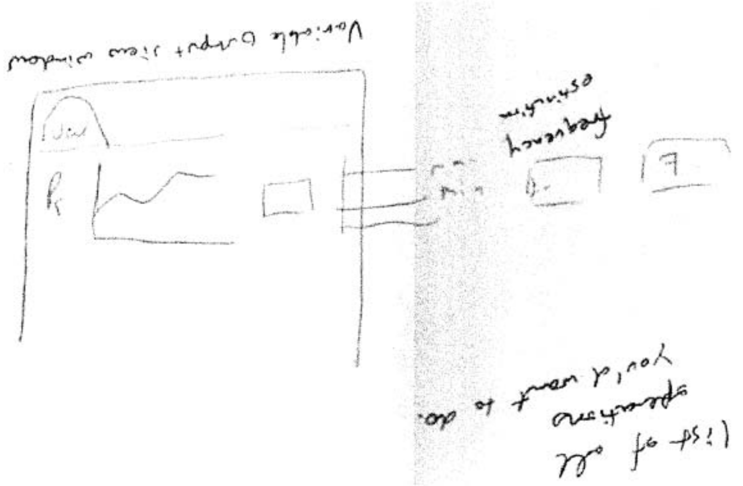


Fig. 2. Communicating ideas visually and verbally at the same time. The participant describes the "Min" and "Max" parts of the sketch as he draws them, to clarify the proposed interaction. Note that the sketch is right-reading the textual commentary is upside-down.

Communicating familiar knowledge, where the original source material is not at hand. In this example, J, a mathematician, uses a sketch to describe a list of parameters that are important to the project. The parameters were identified during requirements gathering, but he wishes to recall everyone's attention to them. He sketches a representation of a colour-coded table to emphasize his point. He creates a very loose sketch of a table, omitting the colour coding, and points to different parts of it as he verbally describes the source material. This looseness characterizes its transient importance. Once the team member was certain that the rest of the group understood him, he stopped embellishing the sketch. This behaviour was common for other sketches in the study.

"B: Well, I haven't seen it. But remember, I was asking, "Do you guys even know what the universe of parameters is, entirely?", about three months ago?

J: It's not worth – actually, I'll sketch it on a piece of paper because that's more in keeping. So, it's a long list of the table with numbers and stuff, and names, and things. And the important thing is, which keeps it really visual is that there are blocks of colour on here for: big red ones meaning, "We need to know this and we haven't done it." Big green ones meaning, "We do need to know this and we've done it." et cetera. And orange ones for, "We've got some idea or something."

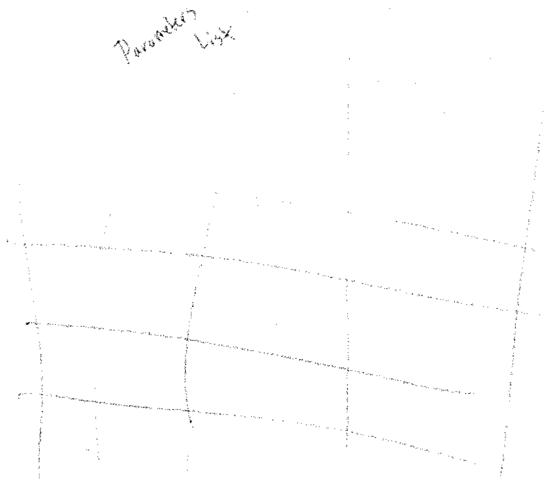


Fig. 3. A quickly rendered sketch referring to knowledge that team members already were familiar with. The participant omitted adding the color that was in the original table, as it was not necessary to make his point.

Confirming understanding of something. In this example, one of the team members explains to a biologist how a particular computational model might be represented by an interactive graph in the proposed interface. Referring to the sketch (Figure 4) he uses it to ensure that his colleague understands how the interactive model will allow for exploration of results that the model will generate.

"P: What we're talking about is a method of testing the models.
M: Testing the models.
P: So, here we have sliders that allow us to adjust these values, even if we an adjust them to ridiculous proportions if we want to and we can still get results out of the model to see what the model does. In exactly the same way we can adjust the model to contain connections just to see what happens – even if those connections we thing in the real world, don't exist. What happens if these were connected? And then that would give us a result. And that would allow us to test things.
J: In this example we know that there isn't such a connection, so it's not a very good example, but what if we weren't sure.
M: But maybe tomorrow they'll discover that it *is there.*
J: In another one, we can have a thing where we have "on" or "off" for a particular kind of connection here. And then we can switch it on or off and see how that affects the graph.
M: Uh-huh.

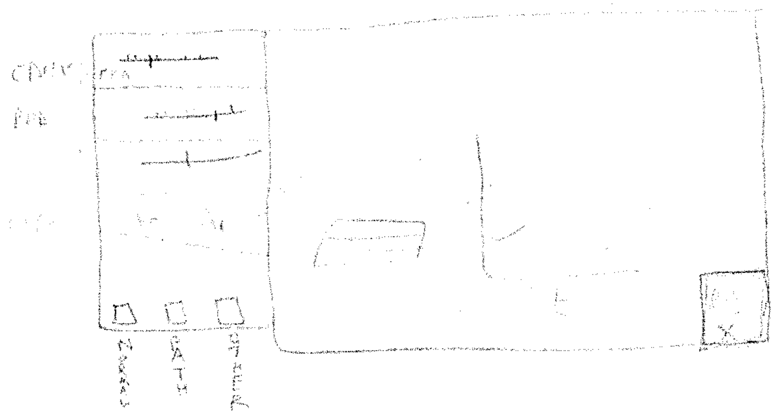


Fig. 4. One participant refers to a sketch to confirm that another team member understands in interaction concept, indicating in the sketch the different objects and their behaviours.

Forming a record of the thought process that can be used later. Participants often used sketches created previously as a point of dialogue for further discussion or for ideation. Here, one person reminds the group of a particular solution that was arrived at earlier. He locates the sketch where that solution had been arrived at and refers to it whilst describing its different attributes, in this case "lumps" and "holes". This particular solution, Model Interface Matching, became a major component of the final specifications. Refer to Figure 5.

"P: Another thing we are looking at here is the difficulty of constructing models and the interfaces those models have with each other and we drew those pictures of the jigsaw pieces?

B: Do we have the pictures?

P: Aha, note for...research: the pictures once again come in useful! We drew these pictures of the pieces and then joined together.

B: [reading from sketch] "Text describing lump."

P: Text describing *hole.* And then the person with the big brain looks at the two and decides whether the *lump* and the *hole* are compatible.

B: Right. What do you call that?

P: It's interface matching. "Model interface matching".

J: Yeah, it is."

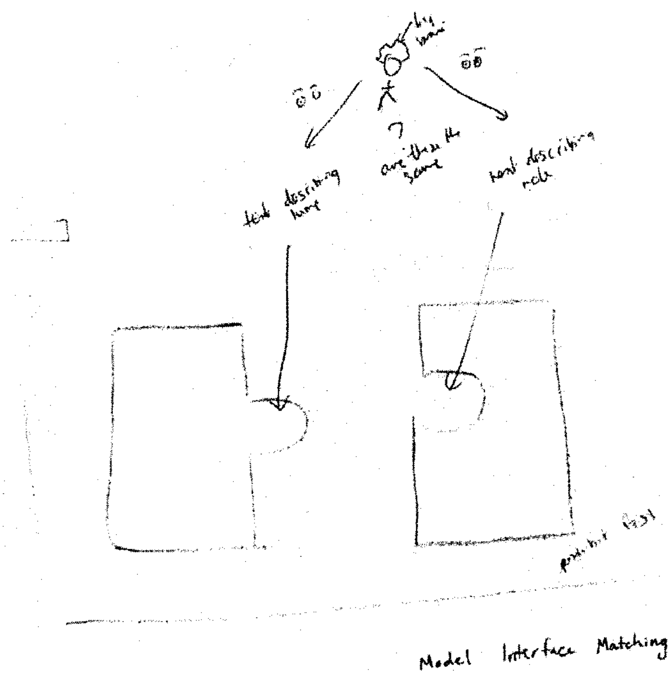


Fig. 5. This sketch formed a record of several decisions about how biologists, represented by the human figure at the top, would use the software to make decisions about connecting mathematical "model interfaces". The sketch was referred to many times over several sessions.

Building complex ideas from simple ones. All of the sketches were contained in a single A3 drawing pad. This sketch history made it easy to refer to previous work and build complicated ideas out of simple components. In this example, members used the previously created jigsaw-puzzle metaphor as a solution for how the software will determine a particular calculation known as waveform relaxation or "WR", depending on user input. The point to note here is that the earlier idea of the jigsaw-puzzle metaphor was available to the participants because they had a written record of it in the sketchpad.

"J: Right, so, the point is that actually because of the mathematics of it, once you start using anywhere, a Oval interface, you have to have Oval interfaces everywhere because you're using the waveform relaxation [WR] algorithm. Or so you might think. But in fact, what you can have is this: when I click – supposing I've got another model here with one of those things, and supposing these are actually separate interfaces – so *that* is an alternative for *that* – I'll colour those in black to represent this.

P: Yes. Good point yes. You can have knobbles that represent – that's quite complicated, really. Yeah, okay.
J: And, I've shaded *that* to say that *that's* matching *that*. Now, when I plug this into here, that disappears and from over here, the computer makes a, one of these –
B: Which is a - ?
J: A thing with a triangular hole and an oval prong emerges from nowhere. The Orchestrator says, "Ah-ha! He's got a mixed-mode WR rate-calculator *thing*, so I'd better cap the rate-calculator – "
P: You've made a transformer? Cool.
B: More than meets the eye."

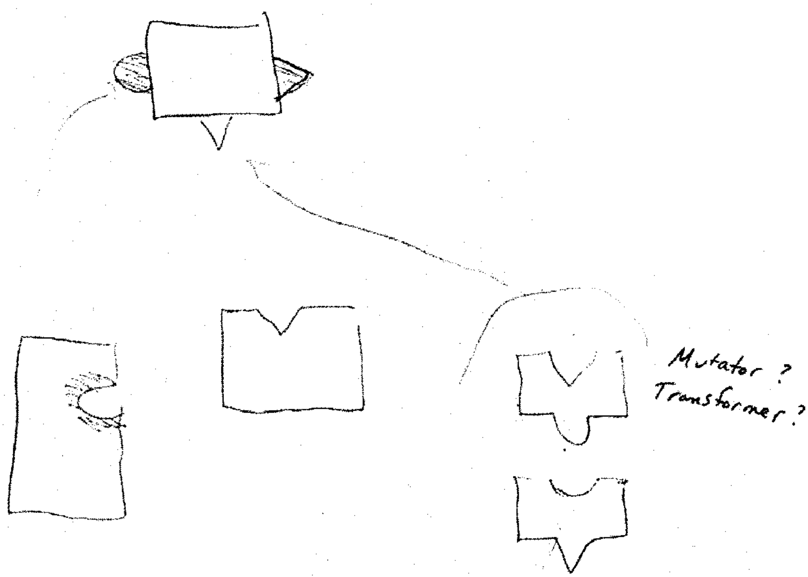


Fig. 6. In this sketch, participants used a previously created puzzle metaphor to build a more complex widget to perform a mathematical calculation called a waveform relaxation.

Explaining difficult concepts. In the domain of computational biology there are naturally many complex concepts. Very frequently it was necessary to reiterate the meaning of these concepts to make sure that everyone in the group had the same understanding. Sketches were often used to augment this process. Here, a team member illustrates that concept of Diet Function for the rest of the group while sketching. See Figure 7.

"J: Right, one of the things we're trying to do with our models is work out how the glucose in your blood changes as you change the amount of food that you eat and the exercise that you do. Diet function, if you like, or food function is, you know, I wake up in the morning and going to make some breakfast...and then I go for a run. And then I have a big lunch and then I kind of sit and don't do very much for a while. So I'm fairly sedentary. And then I have a huge dinner and then lots of beer and then go to sleep, what have you. This thing goes up and down accordingly. We need one of these as the input to our model. And at the moment in order to enter this, it's either numbers in a table or..."

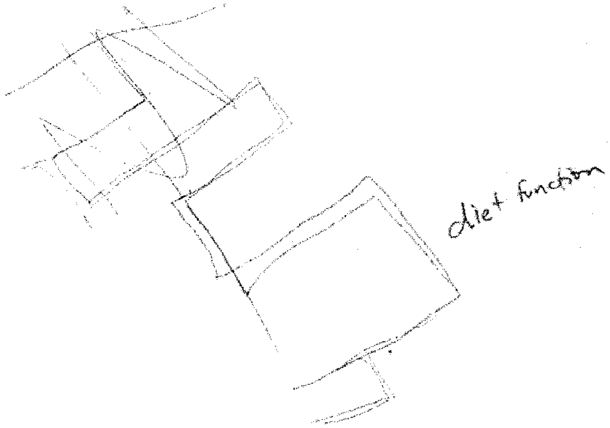


Fig. 7. Participants used sketches to make difficult concepts easier to understand.

2.3.2 Creation

One of the major benefits that sketching brought to the design process was that it allowed people to take ideas in their head and to try them out on paper. This creativity-through-drawing process is well recognised in the fields of graphic communication and architecture [4, 5], but has been little addressed in software engineering. The following three examples from this case study show where how Creation was supported and how it resulted in the emergence of new ideas and capabilities for the software.

Creating abstractions and mnemonics. In the first example, members of the group use sketching to construct their own abstract representations for specific concepts. They create visual mnemonics to encapsulate their ideas. They subsequently used the abstractions as shorthand for explaining the behaviours that the software should support, such as zooming. Essentially, they created the rudiments of a visual language to play with design ideas more effectively. Their Creation helped their Communication. In this example, team members reflect on this activity and how it was helpful:

"J: And we've developed, actually, some mnemonics for different things happening on the computer screen. We've been using what P just drew, a pair of lines with a larger gap at one end than at the other end to mean *zoom*, because that's standard

– you know the thing with the two circles and *that one.* And is there somewhere where we've drawn somebody dragging something?

B: This idea? [pointing to the sketch]

J: Yeah, somewhere or another we've drawn somebody dragging something and drawn an arrow and that kind of thing.

P: There, because that's been dragged over to that window, the "Sandbox" window.

J: So, it may be interesting for future attempts to do this by you guys to teach the clients, us, a vocabulary for drawing computer screens on pieces of paper.

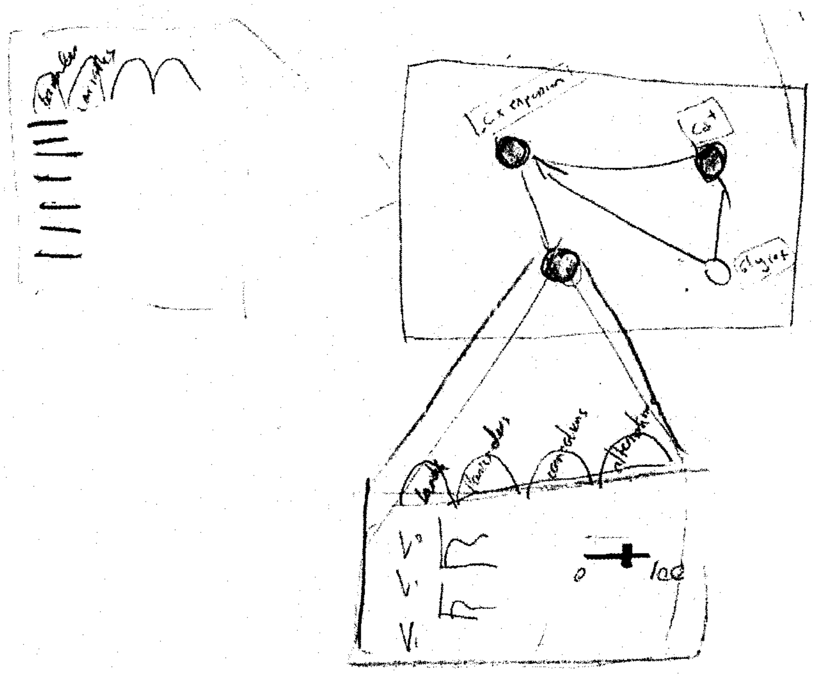


Fig. 8. Use of an abstract representation of zooming to depict the intended interaction of the software. This depiction was subsequently adopted for other parts of the design process and other kinds of zooming.

Figure 8 shows the use of this zooming abstraction, represented by converging lines, as sketched by the participants early in the design process. It is not a novel visual abstraction for the purpose of representing zooming. What is creative here is that the case study participants used the representation as a mnemonic for reminding them how the interface was intended to behave and used this for making future decisions about how the software would work. The also applied this technique in representing different kinds of zooming, both visual and semantic.

Another example is the use of an abstract visual metaphor to constrain user input. One component of the visualisation allows non-mathematicians to construct mathematical models from sub-components. However, the mathematical models in the software have numerous parameters that only interact in certain ways. The designers were looking for a way to force users to construct only valid models out of constituent sub-components but needed to allow users to match interfaces among different models. Through sketching out this design problem, they were able to arrive at a jigsaw-puzzle metaphor that would constrain the user's interaction possibilities. Figure 9 shows an example of this novel representation, which was created by the participants early in the course of exploring this problem.

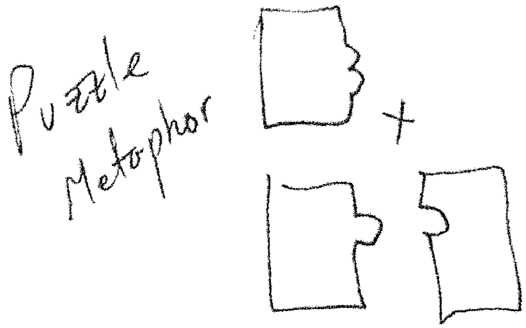


Fig. 9. A visual metaphor for software behaviour.

Creating solutions to problems and novel widgets. In the next example, three participants use a sketch to work out how to solve a problem that the visualisation should show what data ranges are supported by the biology literature. Working together, they create and embellish the novel idea of a slider widget that interactively displays thumbnails of academic papers that support data values in a parameter range. See Figure 10.

"J: I'll just put a little note next to the parameters. One of the things the parameters need to have is: when you drag a slider for a parameter to the point which corresponds to a value with a known supporting text, then it flashes and then it allows you to click and to bring up the paper.

P: That would be really cool, actually! You can imagine a situation where you've got the slider that looks like this and when you're in the middle you've got a list of five papers that support that value and as you get to the end it goes down to one paper and then you go off the end and it says "This is a guess. You're making this number up."

B: That's brilliant yeah. Because then you can know when your stuff is supported and when it's not supported.

J: [Sketching] And you can see, here's a little window. So here's our slidebar. Here's the 'Paper Names' window. Here is – I'm fantasizing now, this would be really hard to implement – a photograph of the cell plate that was used in each of the papers.

P: [Indicating on J's sketch] So when you're here, what you get is this list of five. And when you're *here*, what you get is a much smaller list, you know, two. And then when you're right on the end, then it just says: 'guess'."

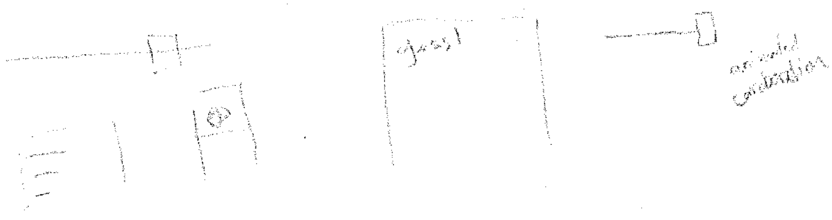


Fig. 10. Creating solutions using novel widgets: the "Provenance Slider".

Sketching also helped participants to overcome the intimidation presented by a stack of requirements specifications with no clear design solution. The prospect of having to overcome a great challenge often creates so-called "writer's block" where designers are unable to begin to work through the design problems. By talking together about each of the design problems and simultaneously sketching out ideas on paper, these blocks were overcome, and creative ideas began to emerge. At the end of the case study, during a reflective discussion, it was the universal opinion of the participants that the sketching helped to overcome inertia and to get the creative process started.

2.3.3 Collaboration

The third major way that sketching facilitated the design process was by helping people to work together. Each of the participants brought his or her unique ideas to the design process, and needed to share them with others. By actively sketching these ideas with others, group input and ideation was made possible. Indeed, there were many points when two or more participants were actively sketching on a common work area together as they played with design ideas. Sketching further supported Collaboration by allowing team members to create scenarios from which to elicit feedback and participation of others. Sketching also triggered understanding among designers which allowed them to build on each other's ideas.

It is important to note that the quality and quantity of collaboration was very significantly affected by the design setting. In this case study, we sought to create a situation conducive to collaboration by ensuring that the environment was comfortable, that the groups were small (no more than 6 people), and that everyone had an opportunity to participate. As facilitator, the UCLIC researcher actively encouraged other team members to try to sketch out ideas for others and supported collaboration in the sketching process.

Creating a scenario and acting it out. There were many occurrences of this. In this example, it is interesting to note that as J creates the scenario, he uses onomatopoeic sounds to emphasize the activity that would occur over time, whilst gesturing on the sketch with his pencil. His portrayal of the scenario conveys to the group what he suggests might happen as users engage with the visualisation. This also allows others to give their input into that behaviour. Refer to Figure 6.

"J: So what happens here is, if I've already plugged those together and then I plug that in, these two bricks come apart and a pair of these [drawing] they go [making onomatopoeic sounds] and these are actually new software entities that the Orchestrator will have to spawn at runtime.

B: And not only are they doing that, but they will have to do that as you did just now, verbally, as an *animated thing*.

J: Of course. Absolutely."

Helping people can build ideas together. In this example, one of the participants, P, explains an idea for a "sandbox" where biologists can perform their own experiments on data. He uses a sketch created in a previous design sessions to describe what activities would be possible in the visualisation, indicating on the sketch as he describes it. He points out different objects in the sketch saying, "this" to help communicate the

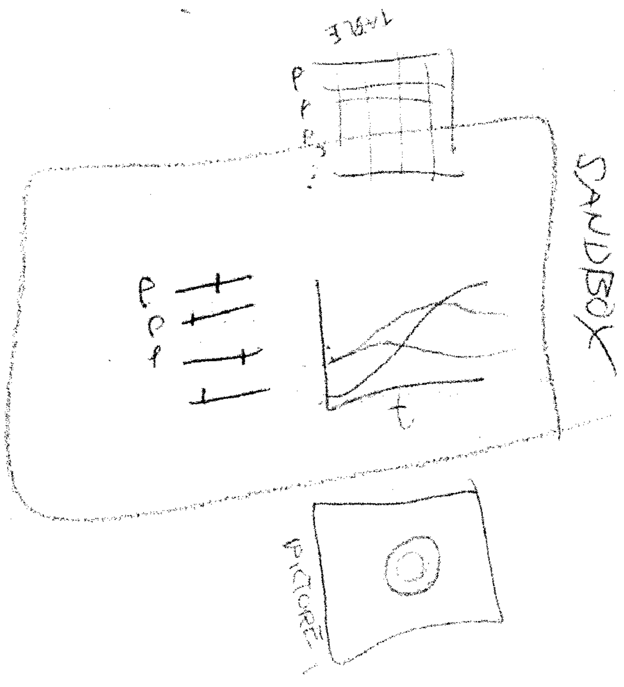


Fig. 11. Facilitating collaboration through sketch sharing.

sandbox idea (Figure 11). This helps biologist M to understand how she might be able to use such a tool and she suggests a possible application. In later design sessions, this allowed her to elaborate on the sandbox, and add her own suggestions for functionality.

P: Finally, we have this idea of a 'sandbox mode'. The idea is, you start off with a blank screen and you say, "Okay in order to look at this really big complex thing that I don't understand, what I really want is *this* parameter here, that I've plucked from over there, and *this* parameter here, and *this* parameter here that I – these are just the ones that I happen to be interested in. And then I want a graph of *this*. And that gives you an interface which just has those things on it and nothing else. And you can add and you can take away from this as much as you want to make it as simple or as complex as you want. If you want to, you can adjust everything.
M: Uh-huh.
M: Like, now I'm going to remove this pathogen, and see what happens if I remove it?
P: Yeah.
M: Or glucagon, or whatever."

Supporting collaborative decisions. Here, the facilitator explores with participants practical limitations of using one visual metaphor over another, by sketching out the alternatives. As a result of expert knowledge of the mathematical constraints, J was able to rule out the use of the puzzle-piece metaphor in Figure 9 and argue for the patch-panel metaphor in Figure 12. By this means, the participants collaboratively identified the better of the two visual metaphors for this application.

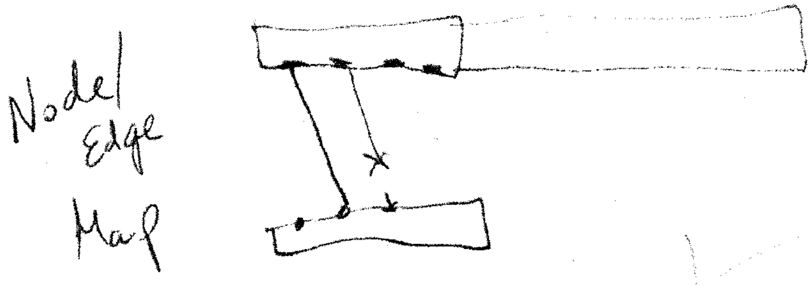


Fig. 12. Choosing among sketched options through collaborative decisions. This "patch-panel" metaphor is a better alternative than the jigsaw puzzle-piece metaphor for the interaction under consideration by the group.

"B: So, here's the puzzle-piece metaphor. Okay another metaphor would be a patch-panel metaphor where you have a piece, [and] a piece here. And they have, essentially, you know, links and receptors.
J: Yeah, this is an old fashioned telephone exchange.

B: It's exactly, the exact metaphor I'm thinking of. And so this visually – you can tell that these two pieces can fit together, whereas these two pieces cannot fit together, right?

P: Uh huh.

B: But the only space you have to work with [within the puzzle piece metaphor] is the perimeter of the visual object in that case. Whereas with these [patch-panels], the perimeter itself doesn't actually encode whether it fits or not, but whether you, say, can draw a link here – it'll either be possible or it won't be possible."

2.4 Outcomes

The final design session included a discussion wherein participants specified what software projects should be produced and prototyped. This was the point at which participants synthesized their knowledge gained during the sketching and design activities and produced a specific set of software specifications. Whereas they had begun the design sessions with no clear solutions in mind, by the end of the process, they were able to articulate exactly what tasks the Information Visualisation software should support, what visual representations should appear on-screen, what interactions should be possible, and how this would support end-users' work processes in computational biology. They then produced a document describing eight sub-projects for which formal specifications could be written and for which code authoring could be started.

3 Discussion

Perhaps the most encouraging result was that the members of the Beacon Project developed eight new software specifications that they did not have before the activity occurred. By encouraging communication, creativity, and collaboration, the use of sketching during the design process allowed participants to evolve new ideas about how the software could solve their most pressing problem on the Beacon Project, namely, allowing both biologists and mathematicians to collaborate in the development of computational models which describe liver function. Moreover, as both biologists and mathematicians were included in the design activities, they were able to more clearly understand the communication challenges among them and to sketch solutions that the software might provide.

Interestingly, as none of the team members were particularly skilled in sketching, they were initially apprehensive and uncertain as to how it would help them to arrive at design solutions. Early in the design activity, the facilitator was more likely to need to prompt participants to the drawing pad and pencils. In contrast, by the end of this process, participants were drawn to use the tools without any encouragement. Their apprehension about sketching had disappeared. People were eager to use the tools to help them communicate and to assist in creating new ideas. They turned to sketching to collaboratively make decisions and rehearse proposed solutions so that colleagues could asses the idea. In discussion during the follow-up interviews, people agreed that they would not have expected this result and that they found it extremely fruitful.

The sketching was particularly effective for helping participants to move ideas from their internal thought processes to a public space where they could be explored and modified. While many participants on the project had privately arrived at good ideas about solutions to specific problems and had perhaps encountered novel solutions their research, these ideas tended to remain "silo-ed". They had not yet shared them with one another on the project. Sketching these ideas together aided in this process and allowed people to modify solutions to more tightly address project needs. Moreover, the participants reported that sketching enhanced their effectiveness in a way that was not possible through verbal dialogue alone. The sketching had prompted changes in thinking about and exploring design problems which would not have otherwise occurred.

These changes in communication, creation, and collaboration are activities that are common to many design disciplines. We believe that these categories of activity would not have emerged so successfully without the use of sketching as an integral part of the design process. What is also novel in this case is that sketching proved uniquely suited to supporting the Information Visualisation design – a highly visual domain. As sketching can support any of these activities at any point, it is appropriate for a generative design process where the novel representational and interaction solutions are unlikely to be known in advance. Also, because they are interdependent, they are non-hierarchical. Each activity supports the others and can occur at any time. Better communication helped the participants in this study to collaborate effectively and stimulated their creativity. Likewise, collaboration improved communication and sharing of ideas. Enhanced creativity resulted in mnemonics and visual that helped to facilitate further communication and collaboration. This non-hierarchical quality is very appropriate for problem solving design activities involving brainstorming and collaboration.

This perhaps highlights what makes the research most promising. Combined with theory development from other areas, further understanding of how sketching supports design in groups may emerge and prove useful to practitioners. While this does not mean that the findings from this case study are generaliseable to other projects in strictly positivist terms, the generic nature of the sketching activity and the successes that are already known to other design disciplines makes it likely that using sketching would be fruitful for others who are engaged in designing highly visual, interactive software. In an area where new theory about design methodology is relatively sparse as compared to more established disciplines such as architecture and graphic communication, any new theoretical knowledge can prove useful to others and can spur further exploration.

4 Future Work

As with any qualitative method, the orientation of this research is to develop theory about the phenomena under scrutiny. This does not necessarily mean that findings will be universally applicable. Yet, we believe that additional research into the effectiveness of sketching will confirm the ways in which it was useful in this case study and yield additional understanding about how it might be helpful to others. We also believe further study would clarify how sketching is related to other low-fidelity prototyping techniques such as paper mock-ups, demonstrate when sketching is most useful, and suggest reasons why. We are optimistic that future practitioners will find sketching useful in to their efforts to surmount Information Visualisation design challenges.

Acknowledgements

We gratefully acknowledge the members of the Beacon Project for their assistance and collaboration in this research. We'd particularly like to thank Peter Saffrey and James Hetherington for their valuable contributions.

References

1. Tufte, E.R. (1983). *The visual display of quantitative information.* Graphics Press: Cheshire, Connecticut.
2. Shneiderman, B. (1994). Dynamic Queries for Visual Information Seeking. *IEEE Software,* 11(6), pp 70-77.
3. Card, S., Mackinlay, J., and Shneiderman, B. (1999). *Readings in Visualization: Using Vision to Think.* Morgan Kaufman: San Francisco, California.
4. Lawson, B. (1997). *How designers think: the design process demystified.* Architectural Press: Oxford, UK.
5. Schön, D.A. (1983). *The Reflective Practitioner: How Professionals Think in Action.* Temple Smith: London.
6. Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison Wesley Longman: Reading, Mass.
7. Baskerville, R. (1999). "Investigating Information Systems with Action Research," Communications of The Association for Information Systems, (19) Article 2.
8. Wilkins, Barry (2003). *MELD: A Pattern Supported Methodology for Visualization Design,* PhD Thesis, University of Birmingham.
9. Alexander, C, et al. (1977). *A Pattern Language: Towns, Buildings, Constructions.* Oxford University Press: New York.
10. Gamma, E., et al. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley: Boston.
11. Borchers, J. (2001). *A Pattern Approach to Interaction Design.* John Wiley & Sons: New York.
12. Preece, J., et al. (1994). *Human-Computer Interaction.* Addison-Wesley: Essex, UK.
13. Hackos, J. and Redish, J. (1998). *User and Task Analysis for Interface Design.* John Wiley & Sons: New York.

Linking requirements specification with interaction design and implementation

Sergio España, Inés Pederiva, José Ignacio Panach, Silvia Abrahão, Óscar Pastor

Department of Information Systems and Computation
Valencia University of Technology
Camino de Vera s/n, 46071 Valencia, España
{sergio.espana,ipederiva, jpanach,sabrahao,opastor}@dsic.upv.es
Phone: +34 96 387 7000, Fax: +34 96 3877359

Abstract: One challenging goal in the context of Software Engineering (SE) and Human-Computer Interaction (HCI) is to provide appropriate bridges between the most well-known software production methods and techniques. SE is supposed to be strong in specifying functional requirements, while HCI is centred on defining user interaction at the appropriate level of abstraction. In any case, general-perspective software production methods that combine most functional-oriented, conventional requirements specification with the most interaction-oriented, user interface modelling are strongly required. In this paper, we present a specific approach in this context, intended to properly combine a sound functional requirements specification with an abstract model of the user interface represented by a CTT model. When the functional specification is enriched with such an interaction model, it is easier to derive the final software implementation that will represent both the structure and behaviour of the system and the user interaction. The presented approach has been successfully implemented in a MDA-based approach called Oliva Nova Model Execution, demonstrating that Conceptual Modeling-based strategies are more powerful when user interaction and system behaviour are modelled within a unified view.

1. Introduction

Conventional software production methods focus on a precise specification of system structure and behaviour. Normally, a class diagram-like model is used to model the system structure, and a process model fixes the functionality that the system is supposed to provide. Many CASE tools [11] [13] [14] have been proposed during the last few years with the objective of providing some kind of automation to manage these data and process models in the context of a well-defined software process.

Generally speaking, when defining such a software process, there is a Requirements Modelling step where requirements elicitation and specification are the issues which are normally based on some kind of use case-based strategy where functional requirements are dealt with. These functional requirements are the source model for creating a Conceptual Schema, where the requirements are reified in the corresponding set of classes and relationships between classes. Finally, it is common

Please use the following format when citing this chapter:

España, S., Pederiva, I., Panach, J. I., Abrahão, S., Pastor, Ó., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 123–133.

to have a of model transformation approach to guide the final step, where the software product that properly represents the initial requirements is provided.

However, from a HCI point of view, these conventional Software Engineering approaches for software production have a mayor problem: to determine how to correctly embed user interaction modelling in such a software production process. It is curious that interaction modelling is not a key issue when requirements and conceptual modelling is faced in a software production process. Together with data and process models, it is very rare to have an Interaction Model playing a basic role at the same abstraction level. It is clear that system structure and behaviour are considered to be basic parts of such a description.

The way in which users interact with the system should be part of the system model in order to be a basic part of the world description and to model the way in which user interaction is going to make feasible the putting into practice the static and dynamic system parts (class architecture and functional requirements respectively) To do this, we claim that functional requirements specification and user interaction design and implementation must be linked in a precise way. Precise means with a formal background, providing a clear set of concepts and a notation to specify user interaction from the very beginning of a software production process; this includes linking the different models involved at the different levels of abstraction starting from functional requirements to a final software product, through the corresponding conceptual schema.

To do this,in this paper, we present a specific software production process, where

1. The first step is a conventional requirements modelling approach, that is based on use-cases, but that has a precise proposal for structuring the relevant functional requirements.
2. When the functional requirements are specified, the corresponding user interaction model associated to the requirements model is specified. To do this with a set of precise concepts and notation and with the required formal support, we use the ConcurTaskTrees (CTT) model [10]. Tasks from the Requirements Model are enriched with the specification of their associated interactions, and a well-defined and contextualized use of the CTT model is proposed to avoid the lack of methodological rules that the use of CTT often suffers from in practice.
3. When both the functional requirements and their associated interaction model have been specified, a Conceptual Schema can be obtained. As an interaction model has a precise semantics provided by the CTT, it is possible to design and implement not only classes and services, but also user interfaces, providing a full software production process from requirements to the final software product where the user interface corresponding to the modelled interaction is properly incorporated.

It is important to note that the Interaction Model that we introduce with the use of CTT complements the functional system specification with the kind of logical interaction design description that the CTT model provides. The functional requirements specification provides the necessary analysis of human life and work context and takes the source Information System as input. The CTT model associates a user interaction to each system task. This will allow the model to execute correctly from the user interaction point of view at the solution space. Within a well-defined software production process, these Interaction Models can be converted into the corresponding sets of user interfaces of the final software product. In this way, system structure, behaviour, and interaction properties are properly dealt with.

Furthermore, this strategy fulfils the current MDA-based approaches that defend the metaphor of model transformation and its automation as the key for defining modern and efficient software processes [4]. In fact, the ideas presented in this paper are already being successfully applied to Oliva Nova Model Execution [1]. This is a model-based code generation tool where our interaction model has been implemented to put into practice all the ideas discussed here.

This paper has the following structure. After the introduction, Section 2 explains how to model functional requirements. To determine how to model user interaction to enrich the Requirements Model, Section 3 describes how the Requirements Model is enriched to incorporate user interaction specification using the CTT model in a precise context for very specific objectives. Once both functional requirements and user interaction are properly specified, Section 4 explains show how to go to the solution space by generating the user interface that implements the user interactions modelled in the previous section. We present an example taken from Oliva Nova Model Execution which is the software generation tat we have used to implement our ideas.

2. Obtaining functional requirements

In the SE field, the first step in building a software system is to capture the functional properties that the system requires. In our software production process, this is done through the definition of a Requirements Model [2] [3]. This model contains a description of the objectives and external behaviour of the system, that is, *what* the system must do without describing *how* to do it.

The Requirements Model incorporates a set of complementary techniques: the mission statement, the functions refinement tree, and the Use-Case Model.

The *Mission Statement* is a high-level description of the nature and purpose of the system. Through this definition, it is possible to accurately determine what the system will and will not do.

The *Functions Refinement Tree (FRT)* represents the hierarchical decomposition of the business functions of a system independently of the current system structure. The resultant tree is merely an organization of external functions and does not say anything about the internal decomposition of the system. The leaves of this tree represent the functions of the desired system, which are the use cases. This gives the entry point for building the Use-Case Model instead of starting from scratch, and it avoids the potential problem of mixing the abstraction levels of use cases.

Once we have defined the *FRT*, the next step is to create the *Use-Case Model.* A use case is an interaction between the system and an external entity. This interaction can usually be decomposed into an activity set (Case Use specification) defined at this level as atomic functions. The leaf nodes of the FRT (elemental functions) are considered to be primary use cases; they represent the most important functions of the system. It is also possible to have secondary use cases. These use cases are scenarios that have no direct correspondence to the FRT; however, they are important for organizing and managing complexity through relationships among use cases that are stereotyped as EXTEND and INCLUDE.

In order to explain our proposal, we use an example taken from a real system from the Oliva Nova Model Execution portfolio: the Bullent's Water application. This system is used in a company that delivers water to homes. The main functions of the system are: read client's meter, emit an invoice, register the use of some material in a repair, and maintain the stock in the warehouses. For the sake of simplicity, we have centred our attention only in one task of this system: the task to *create a new subscriber* in the company. This task is composed by several subtasks: create a client, create a transfer, create a meter, create a new meter address, create a destination and verify the new subscriber data. Figure 1 shows the functions refinement tree for the task being studied. It contains all the functions needed to create a new subscriber.

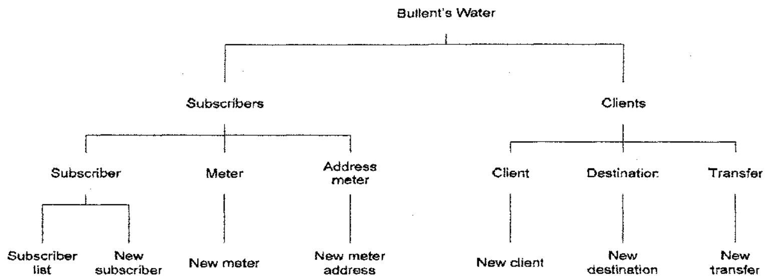
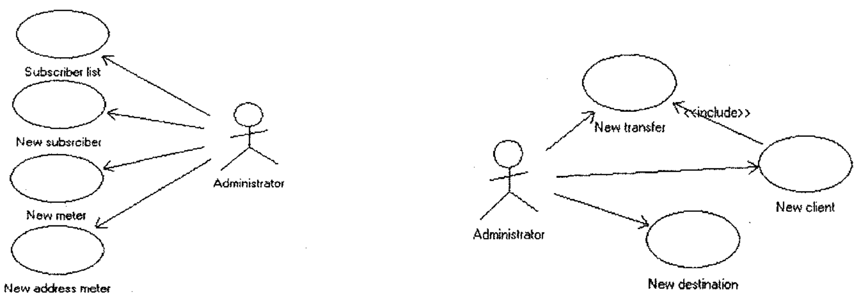


Fig. 29. Function refinement tree

The next step is the definition of a Use-Case Model. To draw this diagram, we have used RETO[2] (Requirements Engineering TOol).



Subscriber functional group (a) Clients functional group (b)

Fig. 30. Use Case Diagrams

These diagrams show all the functions that make up the task "create a new subscriber". There is an *include* relationship between the "New subscriber" use case

[2] http://reto.dsic.upv.es/reto/

and all the other use cases, with the exception of the "New transfer" use case. These lines have not been drawn to simplify the diagram reading.

Finally, all the steps of the use-case behaviour must be specified. As an example, the "New subscriber" use case is detailed in the following template:

- **Description:** It creates a subscriber in the company. It records information about the person, his/her meter, his/her destination and his/her transfer.
- **Steps:**

1. Create a new client INCLUDE USE CASE: New client
   1.1 (Inside New client) Create a transfer for the new client INCLUDE USE CASE: New transfer
2. Create a meter INCLUDE USE CASE: New meter
3. Create a meter address INCLUDE USE CASE: New meter address
4. Select a zone
5. Select a category
6. Select a house type
7. Introduce dispenser diameter
8. Introduce branch number
9. Select discharge date
10. Create a destination INCLUDE USE CASE: New destination
11. Introduce connection length
12. Introduce connection diameter
13. Select invoice option
14. Select invoice type
15. Introduce concept
16. Select currency
17. Save the new subscriber

As a consequence, all system functional requirements are captured in the Requirements Model. The leaves of the functions refinement tree are tasks functionally described using the use case templates. However, the next step is to properly capture the user interaction.

3. Modelling interaction with CTT task model

After using the RETO tool to elicit and document system requirements, we have a complete decomposition of the system behaviour. As the leaves of the FRT are well-defined use cases, the functional requirements of the Information System are properly stated; however, the interaction between the user and the system is still not sufficiently documented. To be able to use this functionality, this section presents a way to deal with interaction requirements. It not only shows how to model an interaction but also to reason about it. To do this, we use the ConcurTaskTree notation (CTT) in the framework of our development process [9] [10].

CTT, as originally proposed by Paternò, constitutes a formal notation to express human-computer interaction as a graphical task decomposition. The main constructors are tasks and relationships between them. Tasks can be of four types (abstract, interaction, application and user task) and several relationships are available to link sibling tasks. These basic blocks for building the task model were well defined, but little criteria has been provided for its application.. Thus, as the HCI and SE communities started using this notation, different approaches were used, which has resulted in creating its own criteria for task decomposition and granularity. We define the use of the CTT notation according to our specific context.

One of the first criterion needed is the starting point for the interaction specification: we are not building a single tree that describes all the interactions between the user and the system, but rather a forest of task trees with the root of each tree being a leaf of the FRT, that is, a use case of our functional Requirements Model. Given the correspondence between use cases and the upper abstract task node, we decompose these into a *data introduction* task followed by an *application* task that processes the data entered by the user (see Figure 3). For some complex use cases, the interaction is fragmented into several subtasks which are embedded in a similar structure.

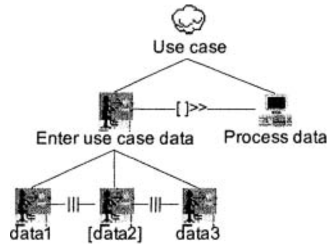


Fig.3. Example of the interaction related to a generic use case

The decomposition is continued until the individual data elements that make up the message that is being communicated to the software system are reached. These data elements are commonly derived from (and consequently mapped to) the fields of business forms; they are documented in the use case description (see example in Section 2) and will later be mapped to data entry interface fields. This way, we can model the edition of the message that the user wants to communicate to the system.

We also include application tasks, which correspond to the processing of the information supplied by the user and the feedback resulting from this process. These tasks are triggered by the user with an implicit end-of-edition signal (e.g. after the last piece of data is entered) or an explicit end-of-edition signal (e.g. the user presses a button). Application tasks are later mapped to services of objects or global transactions; in the latter case, we could decompose the application task into sub-tasks that would model the structure of services involved in the transaction. However in order to keep the task tree as simple as possible, this refinement would be done in a separate tree in a packet-like approach.

Figure 4 shows the CTT for the *Create subscriber* use case that was described in the previous section.

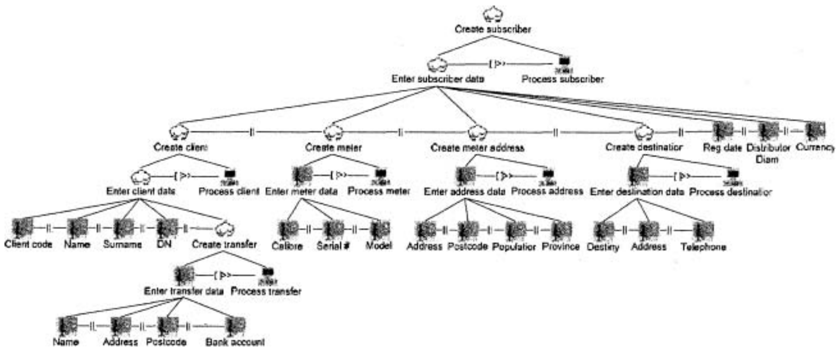


Fig. 4. CTT for the *Create subscriber* use case

Since a use case that includes other use cases a complex one, the resulting CTT has nested abstract tasks that also consist of data entry and process; for example, the need for creating a new meter to be related to the subscriber.

It can be seen how the abstract task in the root maps to a use case of our functional Requirements Model and also how the decomposition stops at the level of individual data elements. For the sake of simplicity, we have limited the number of data elements in Figure 4 by informally using ellipsis points.

4. Generating the user interface

The previous sections define ways to specify the requirements of a software system in terms of its functionality and user interaction. These requirements describe what the system needs to provide and does not describe how to implement the solution. In order to complete the software development process, the software should be implemented. For this phase, there are tools that provide support for designing conceptual models derived from the requirements and that generate the source code of the designed application.

Although there are many popular tools that support this process [11] [13] [14], none of them consider the design of the user interface. Therefore, none of these tools provides automatic derivations for the final user interface. The above mentioned OlivaNova™ Model Execution (ONME) implements the method proposed in OO-Method [8] which provides both the automatic transformation desired and also takes into account the abstract design of user interfaces.

In accordance with the directives of the Model-Driven Architecture (MDA) framework [4], ONME has a direct correspondence with the different models that MDA proposes. OlivaNova is a MDA-based technology that implements model transformations in an industrial context [6]; specifically, it allows the automatic generation of complete applications from the Platform-Independent Model, which describes the information system at the analysis level and comes right after the Computation-Independent Model.

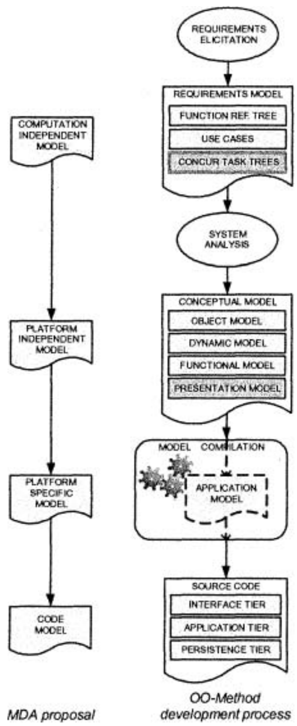


Fig. 5. Usage of ConcurTaskTrees in a MDA-based development framework

In the Platform-Independent Model level, OO-Method includes the *Presentation Model* as a view of the *Conceptual Model*, which is a model for the abstract specification of user interfaces. Based on a pattern language called Just UI [5] [7], this model allows the analyst or the designer to manipulate, configure and link those patterns in order to build the interface abstract model.

With the complete design, the source code of the final application can be obtained with OMNE, resulting in a three-layer application which includes the application logic, the database, and the user interface.

The design and development process can be resolved based on the requirements obtained and the interaction designed, but there is no explicit link between them. Although there is no direct binding, we have found a relationship between the CTT model and the Presentation Model and its final developed interfaces.

As experimentation, we designed the Presentation Model following the design of the CTT task model using the patterns provided by Just UI [5] [7] and we obtained the resulting interfaces. Figure 6 shows the final window resulting from the application of this technique with the CTT previously described in Figure 4.

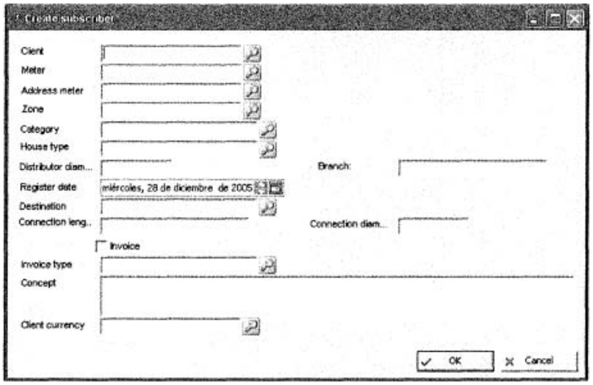


Fig. 6. Generated window

In these experiments, we found that there was a service behind each task. Therefore we defined a window for each task. to the comments of each window corresponded to the definition of the task tree and the following rules were applied in the transformation:

1. Each data introduction had a corresponding widget, corresponding to the data type that the system required.
2. Each *application* task had its corresponding Ok button in order to apply the changes
3. Each *abstract* task called another window to fulfil the entry (obviously, the window called was generated by the decomposition of the abstract task).

Figure 7, presents a graphical explanation of these correspondences.

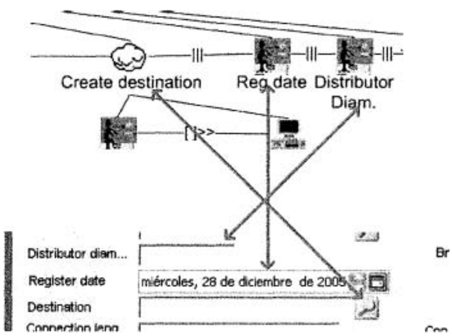


Fig. 7. Correspondences between CTT and the final widgets

The developed application is currently in production phase in the company. The final users have indicated that they are happy with the user interface. The results of this work suggest that we can continue working in this direction in order to link the specification of the CTT task model and the Presentation Model. The mapping of CTT primitives and the conceptual patterns proposed by Just UI could open up a new avenue for automatically generating user interfaces based on the CTT task model with a guaranteed valid semantic granted by CTT task models.

5. Conclusions

5.1 Current research

Current software production methods are characterized by not having the design of user interaction as part of the system lifecycle. As a result, most of the problems in software development are related to user-interface design. Similarly, current UI tools in the HCI field tend to focus on issues such as colours, fonts, and alignment, which are more appropriate in the later design stages. In addition, most of these tools do not support a process that links UI design with system behaviour. What is needed is a complete software production process that properly integrates system functionality, behaviour, and user interaction in the early stages of the system lifecycle. This process should also allow the sketching, modelling, and prototyping of UIs. It is our position that SE and HCI software production methods and techniques can be integrated to provide such an approach.

In accordance with these ideas, in this paper we have presented a software production process that integrates model-based and task-based approaches to user interface design. This process properly combines a functional requirements specification with a hierarchical model of the user interface represented by a CTT model. In fact, the system functionality and behaviour as well as the user interface are modelled at an early stage of the system lifecycle. This is done thanks to the Oliva Nova Model Execution approach which allows the modelling and automatic generation of software applications. This is a pure model-transformation process where conceptual primitives of the Conceptual Model level are converted into their associated software component counterparts, within what we might call a Conceptual Model Compilation Process. This approach can help to bridge the gap between SE and HCI since it addresses the issue of modelling user interaction, which has often been ignored by SE researchers.

5.2 Pending Issues

Although the use of CTT task models in the context of the OlivaNova Model Execution approach seems to be useful, we must still verify this empirically. We are planning to run an empirical study to assess the effectiveness and perception of the analysts in the use of our software production process with and without CTT task models. The goal is to verify the quality of the user interfaces obtained following a well-defined semantic provided by the CTT task models.

In addition; we want to make an in-depth analysis of the use of MDA-based techniques to provide a full software production process. In this process, the unified modelling of system structure, system behaviour, and user interaction should guide the different model transformations at the different levels of abstractions from requirements to the final software product through the corresponding conceptual schema. The final goal is to have a General Model Compiler as the main software production tool, which will make it easier to guarantee that the final software product is the correct representation of the initial user requirements. All of this should be based on a common, sound, and rigorous engineering approach for dealing with the software production process as a whole, including static, dynamics and interaction.

References

[1] Care Technologies: http://www.care-t.com Last visited: Dic-2005

[2] Insfrán E., Pastor O. and Wieringa R., Requirements Engineering-Based Conceptual Modelling. Requirements Engineering 7 (2): 61-72 (2002).

[3] Insfrán E., Molina P., Martí S., Pelechado V., Requirements Engineering applied to the Conceptual Modeling of User Interfaces, IV Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS'2001), Santo Domingo, Heredia, Costa Rica, pp. 181-192, April 2001 (in Spanish)

[4] MDA: http://www.omg.org/mda Last visit: Dec-2005

[5] Molina P., Specification of User Interfaces: from requirements to automatic generation, PhD Thesis, Dept. of Information Systems and Computation. Valencia University of Technology , March 2003 (in Spanish).

[6] Molina, J.C., and O. Pastor. "MDA, OO-Method and the OlivaNova Model Execution technology". I Workshop on model driven development, MDA and applications. Málaga, Spain, 2004.

[7] Molina, P. J. Meliá, S. and Pastor, O. (2002), Just-UI: A User Interface Specification Model. In Ch. Kolski and J. Vanderdonckt (Eds.), Computer-Aided Design of User Interfaces III, Kluwer Academics Publisher, 63–74.

[8] Pastor, O., Gómez, J., Insfrán, E. Pelechano, V. (2001) The OO-Method Approach for Information Systems Modelling: From Object-Oriented Conceptual Modeling to Automated Programming. Information Systems, 26(7) 507–534.

[9] Paternò, F. (2000). Model-Based Design and Evaluation of Interactive Applications, Springer-Verlag, Berlin, Alemania.

[10] Paternò, F., C. Mancini, et al. (1997). ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction, Chapman \& Hall, Ltd.: 362-369.

[11] Posiedon http://www.gentleware.com Last visited: Dic-2005

[12] Pribeanu, C. and J. Vanderdonckt (2002). "A methodological approach to task-based design of user interfaces." Studies in Informatics and Control 11(2): 145-158.

[13] Rational http://www-306.ibm.com/software/rational/ Last visited: Dic-2005

[14] Together http://www.borland.com/together Last visited: Dic-2005

Human Motion Analysis in Treadle Pump Devices

C. Pereira[1], J. Malça[2], M. C. Gaspar[3], and F. Ventura[4]

[1] Departamento de Engenharia Mecânica,
Instituto Superior de Engenharia de Coimbra,
Rua Pedro Nunes, 3030 - 199 Coimbra, Portugal.
candida@mail.isec.pt
[2] Departamento de Engenharia Mecânica,
Instituto Superior de Engenharia de Coimbra,
Rua Pedro Nunes, 3030 - 199 Coimbra, Portugal.
jmalca@mail.isec.pt
[3] Departamento de Engenharia Industrial,
Escola Superior de Tecnologia, Instituto Politécnico de Castelo Branco,
Av. do Empresário, 6000-767 Castelo Branco, Portugal.
calvete@est.ipcb.pt
[4] Departamento de Engenharia Mecânica,
Faculdade de Ciências e Tecnologia da Universidade de Coimbra,
Pinhal de Marrocos, 3030-201 Coimbra, Portugal
fernando.ventura@dem.uc.pt

Abstract. Poverty and hunger are common problems in developing countries where agriculture is seriously affected by lacking of irrigated land. The treadle pump is an effective low cost device, which combines higher water discharge rates with ease of operation. Improving the performance of the treadle pump, considering dimensional and structural requirements, manufacturing and maintenance aspects, cost reduction and ergonomics is the aim of the authors. A human centered approach is proposed to enhance the performance of these pumps, firstly because the user's influence on the treadle pump's design has not been completely analyzed so far and secondly because water discharge depends significantly on the user's performance. A parametric study was carried out. It was found that a comfortable pumping position requires feet angular positions between -10° and +10° and treadles must be large enough to allow different pumping positions and operator's height. A new numerical approach is proposed for modeling the user's movement.

1 Introduction

Poverty associated to hunger are among the greatest problems in the world mainly in developing countries of Asia and Africa, where approximately one third of the population endure of chronic subnutrition. The agriculture is seriously affected in these regions where only 4% of arable land is irrigated. Thus, finding sufficient water for irrigation is one of the principle ways to increase production in agriculture and consequently one of the major challenges to improve food security and family incomes. In order to boost the production capacity of the typical farmer in Africa, lower-cost, water-efficient irrigation equipment is crucial. Among the most exciting innovations in manual irrigation technologies is the treadle pump [1,2]. The treadle

Please use the following format when citing this chapter:

Pereira, C., Malca, J., Gaspar, M. C., Ventura, F., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 135–145.

pump is an effective low cost alternative to the rope and bucket system as it combines higher water discharge rates with ease of operation. Several beneficial impacts arise from the adoption of the treadle pump: *i)* improvement of family nutrition; *ii)* benefits on farming practices – increased agricultural area under irrigation, additional and new crops grown each season, reduced irrigation time, less tiring irrigation work – and *iii)* benefits at the economic level – high family incomes due to marketing of vegetables and employment opportunities for the whole supply chain of manufacturers, retailers and selling agents of treadle pumps. In this way, the treadle pump puts irrigation at the service of the poor.

A treadle pump comprises twin cylinders fitted with pistons that rise and fall when an operator, standing on the treadles connected to the pistons shifts his body weight side to side in a walking motion. In this manner, a suction pressure is created inside the cylinders and water is forced to enter the pump cylinders through the inlet pipe connected to the junction box. A non-return valve is fitted to the end of this pipe, allowing water to enter the pipe and preventing it from flowing back to the water source. Several designs of this device are currently available, all having the common basic components [3] as presented in figure 1.

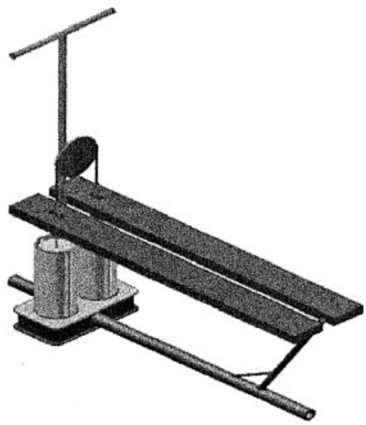


Fig. 31. The treadle pump

Different studies were developed comparing the performance of the treadle pump vis-à-vis other water-lifting options [4,5]. These studies showed that the treadle pump is easier to operate than other types of manual irrigation systems and its efficiency is also greater. Compared to other manual lifting devices, treadle pumps use human power in a relatively efficient way [6]. The pump employs the user's body weight and leg muscles, being much less tiring than other manual pumps that use the upper body and arm muscles [7]. Changing the driving power from arms and hands to feet and legs has been one of the most important innovations brought by the treadle pump.

The principles of human powered pumping systems depend mainly on the binomial operator-pumping device and must be completely understood to provide practical constructive solutions in order to enhance their performance. Thus, many features must be studied and enhanced so a multidisciplinary methodology focusing dimensional and structural requirements, manufacturing and maintenance aspects, cost reduction as well as ergonomic aspects is expected to obtain significant results in the treadle pump enhancement. This methodology is summarized in figure 2.

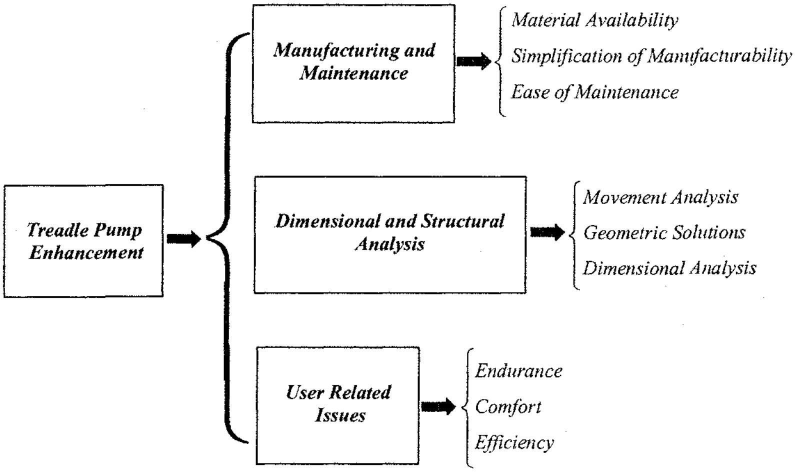


Fig. 32. Treadle pump enhancement methodology: key aspects

In this work, a human centered approach (user related issues) is proposed by the authors to enhance the performance of these manual pumping devices, firstly because the influence of the user on the treadle pump's design and constructive solutions has not been completely analyzed so far and secondly because the water amount pumped on this type of device depends significantly on the user and its performance. Hence, aspects like endurance, comfort and efficiency must be studied when considering the treadle pump's enhancement. The analysis of the treadle pump's ergonomic aspects is expected to help deriving new effective design directives. The gait and posture of the treadle pump's user, and in special the feet position, play a significant role on the performance of these human powered water lifting devices. In addition, the user's weight and physical shape are also essential, as the treadle pump's efficiency depends mainly on its operator's performance. Due to its operating simplicity, this device allows users like children and women [3,6,7,8]. Hence, the user's anthropometric influence must be carefully analyzed, which is the aim of the forthcoming section.

2 General considerations

Persons using treadle pumps intend to elevate the maximum water volume within the shortest time. The pump and its mechanism must be efficient, however they are moved by human power and, therefore, persons are fundamental. Pumps must be actuated for long periods, so comfort of its user is essential, otherwise he will get tired rapidly. Besides, pumps can be used by males, females or children. Figure 3 illustrates

Fig. 33. Operator's position on the treadle pump

the posture of a man on the treadle pump. Ergonomics is the science that promotes adaptation of human and material resources for the best performance. This requires a deep knowledge of the human body in terms of anthropometry and comfort. Anthropometry is the discipline of design that studies human body sizes and it is strongly connected to ergonomy. The main aspects of the interaction between ergonomics and anthropometry are the sizes of human body segments, muscular forces, posture, and movements, since they are intimately related with comfort, safety and functionality.

Data on human body sizes vary from person to person and from country to country. They also vary from youth to eldness and with historical time. Height of Portuguese people has increased since antiguity and particularly during the XX century as a result of improvements in food and life conditions. Modern societies have recognized the importance of anthropometry and technical data is already available. However, in developing countries this is not a central issue and data is scarse.

The human body can be considered as a group of rigid segments (the bones) moved by muscles. Muscular fatigue is the decay in the capacity to generate muscular tension and occurs due to a sub-critical extended activity, or due to a high intensity and low duration activity. In a sub-critical exercise fatigue is a consequence of reduction in glycogen stocked in the muscles, even if oxygen is available. During prolonged exercise water ingestion and feeding are also fundamental. The fatigue of the human body is quickly reached when an isolate number of muscles is used or when the body is forced to work in an uncomfortable anatomic position.

3 Ergonomics of the Treadle Pump

The performance of the treadle pump depends on the interaction of the triad operator-pump-mechanism as illustrated in figure 4.

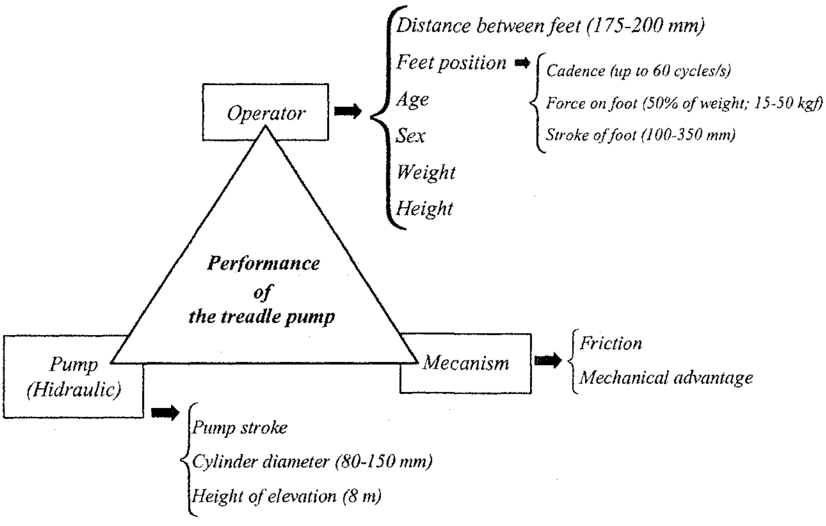


Fig. 34. Parameters determining the treadle pump performance

As referred above, ergonomic aspects are crucial when the aim is to improve treadle pump performance. Therefore, geometry and size of the different pump components should be designed and selected to guarantee the operator comfort and thus optimize the human power during large operating times. Unfortunately, few works can be found in the literature concerning this key aspect of the treadle pump. In 1993, Thomas [8] presented a definition of ergonomic efficiency as the relation between the (real) energy effectively transmitted by the operator to the treadles and the energy transmitted in ideal operation conditions. However, the parameters involved in this relation are difficult to quantify. Srinivas and Jalajakshi [5] evaluated the operacionality and ergonomic behavior of these devices inquiring several users. The results showed that the majority of the pump operators did not feel physical tiring in excess; nevertheless, the operating times were short. In fact, ergonomics is one dimension of the investigation where improvements are promising [3] and needed.

The treadle pump as a human powered device uses the operator's body weight and employs the most powerful muscles of the human body –leg muscles– instead of power from arms and hands to lift water from different sources. However, the use of these devices for long periods –up to 8 hours a day– requires physical strength, thus confort is crucial. So, in order to perform the optimization of this type of devices the first step is to study the issues related with the user.

The operator can be a male, female or child, as referred above, hence the parameters related with the operator can vary significantly, namely weight, height and physical shape. The treadle is a flexible machine allowing easy adaptation of different person, even children. The operator can select the foot position in relation to the pivot point (see figure 5), which modifies the cadence and the force on foot. Lower cadence corresponds to a higher force and lower stroke, which is adequate for heavier persons. On the other hand higher cadence, higher stroke and lower force are adequate to young people. Recommendations can be found in the literature for cadences lower than 60 cycles/min and foot strokes between 100-350 mm [3,8].

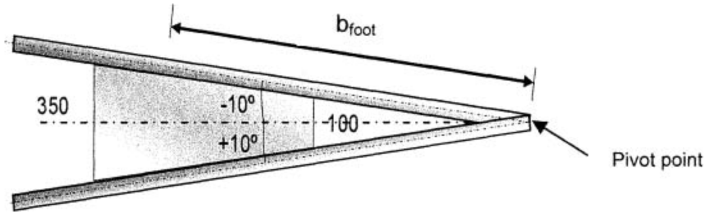


Fig. 35. Feet extreme positions related to the pivot point of treadles

The leg's movement is similar to the combination of step climbing and walking, so it is easily acquired and repeated sequentially in a mechanized way. However, the fatigue of human body is quickly reached when an isolate number of muscles are used or when the body is forced to work in an uncomfortable anatomic position. Furthermore, and for long periods, the force transmitted by the user to the treadle must not exceed 50% of his weight. For short periods this limit can be increased up to 70%. Since the treadle pump can be used by different types of operators (males, females and children) it should be designed for resistance forces between 15 and 50 kgf [3,8].

The gait and posture of the treadle pump's user, and particularly the feet position, play a significant role on the efficiency of pumping. Figure 6 represents the extreme angular positions of feet. The equipment should be designed in order to guarantee that the mean position of operation matches a neutral angle of θ=0°–horizontal position– since in this position the foot force is maximum because the force acts vertically [9]. However, a comfortable pumping posture requires feet angular positions between -10° and +10°. These values depend on the cylinder's position along the lever and on the cylinder's stroke [10]. Considering that the foot stroke should vary from 100 to 350 mm and θ between -10° a +10°, as illustrated in figure 5, then the distance from foot to the pivot point –b_{foot}– can vary between 280 and 1000 mm.

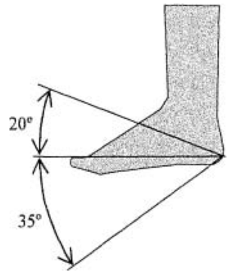


Fig. 36. Extreme angular positions of feet

The angular position of the legs depends on feet distance and represents another important ergonomic parameter (figure 7). In fact, treadles must be large enough to be suitable for different operator's height and allow selection of the best position. Additionally, relatively large treadles allow flexibility for the angular position of feet (see figure 8). On the other hand, enlargement of feet supports increases weight, with a negative impact in terms of pump's portability, which is another key design parameter.

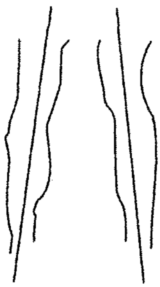


Fig. 37. Legs angular positions

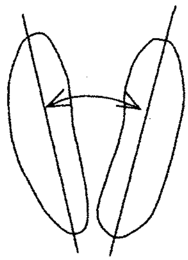


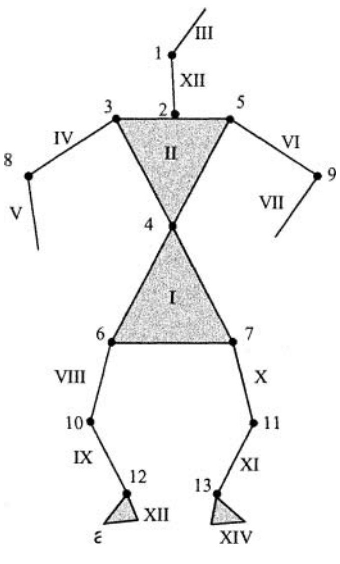
Fig. 38. Feet angular positions

4 User's movement modelling:"The puppeteer model"

As a future work, authors propose the development of numerical and experimental procedures to study the human body movement in a treadle pump. The interest in the analysis and simulation of different human actions stems from the need to predict with sufficient accuracy the mechanical behavior of the human body in various conditions of activity [10]. Human tracking plays an important role in this movement analysis, as it intends to recover continuously the global positions of the human body when in motion. The information thereby acquired can be used in many fields, such as virtual reality [11,12], effect of industrial environments and activities [13,14], medicine (clinical study on orthopedic patients and rehabilitation) [15,16], sports (sport performance analysis and athletic training) [17,18] or design of ergonomic equipment. Numerical studies of human body movement involve two main tasks: reconstructing the body and moving it. This is described in the next section.

4.1 Body Modelling

Different segments are usually considered, each representing a portion of the human body. A variable number of segments can be found in the literature. Ayoub *et al.* [13] have considered 5 segments in a 2D analysis. Ning *et al.* [19] have considered 14 rigid bodies –upper torso, lower torso, neck, two upper legs, two lower arms, two thighs, two legs, two feet and a head–, which are made of bone and soft tissues. Bone segments are considered non-deformable and therefore are represented as rigid bodies. Soft tissues may or may not be considered deformable. This last option is widely accepted in the literature, being the whole body segments regarded as rigid bodies. Tissue deformation may have inertial effects on movement kinetics during highly accelerated movements. Different anthropometric data is needed to reconstruct the body, namely distances, masses and inertia moments. This data varies significantly from person to person, which makes difficult its definition. Figure 9.a presents the human biomechanical model considered in the present work, based on the model proposed by Silva *et al* [10].



Body no.	*50% Human Male* L_i *[m]*	*50% Human Male Mass [Kg]* *
I	0.260	14.2
II	0.250	24.9
III	0.230	4.24
IV	0.320	1.99
V	0.260	1.84
VI	0.320	1.99
VII	0.260	1.84
VIII	0.410	9.84
IX	0.385	4.81
X	0.410	9.84
XI	0.385	4.81
XII	0.160	1.06
XIII	0.053	1.62**
XIV	0.053	1.62 **
$L_{3\text{-}5}$ *(a)*	0.375	-
$L_{6\text{-}7}$ *(a)*	0.188	-
$L_{1\text{-}2}$ *(b)*	0.199	-
$L_{1\text{-}3}$ *(b)*	0.155	-

Fig. 39. a) Human biomechanical model; b) Mass and dimensions of rigid bodies (*- according to Silva et al. [10]; **- estimated values)

It has 14 rigid segments (2 for the head, 2 for the trunk and 10 for the members) and 13 kinematic joints. The total number of Cartesian coordinates is therefore n=14x6=84. Bodies IV-XI were assumed to have no rotation about their longitudinal axis (8 constraints). Joints 8, 9 (elbows), 10, 11 (knees) were assumed to be revolute joints (4x5=20 constraints). Joints 1, 2, 3, 4, 5, 6, 7, 12, 13, were assumed to be spherical (9x3=27 constraints). The mobility of the body is therefore k=n-m=84-55=29 degrees of freedom. The data needed to implement the model comprises the mass, dimension, principal moments of inertia and center of mass location for each body. The table presented in figure 9.b includes mass and dimension data considered for each body segment. These parameters were used in motion computer modeling, using commercially available rigid multibody software. Figure 10 shows the biomechanical model as modeled in MSC.visualNastran.

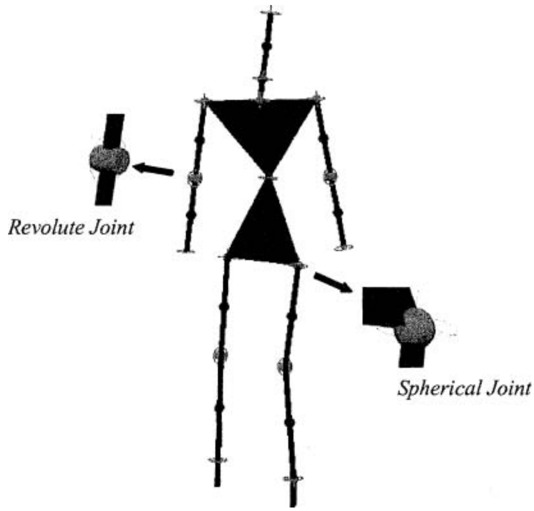


Fig. 40. Biomechanical model obtained using MSC.visualNastran software

4.2 Body Movement

Stereophotogrammetry methods are used to reconstruct 3-D landmark coordinates from photographs, radiographs and video images. Video based optoelectric systems are nowadays the most popular in movement analysis. These systems are used to track, by means of cameras, the 3-D position of a set of points. This allows a non-invasive estimation of the instantaneous position of points in a 3D measurement volume. Based in this concept, an experimental setup is being developed to track the movement of the operator in a treadle pump. Instantaneous positions of markers will be determined from a collection of digital images obtained by video cameras.

The rigid bodies that integrate the biomechanical model will be controlled (moved) by external actuators displacement/length type (also implemented in MSC.visualNastran software), instead of muscles, moving strategically the marker points of the model according to the previously tracked positions. Considering the similarity of this approach with the puppeteer control of a moved dummy, this procedure is named *puppeteer model*. The length of each actuator in each instant is obtained from experimental data (for each actuator is of time *vs.* length). The biomechanical model associated with the treadle pump parametric model, illustrated in figure 11, can be used to study the movement of the centre of mass during each cycle and to calculate the joint reactions and the resulting forces at the user's feet, in order to improve the performance of the binomial treadle pump/user. This is the next step of this research work.

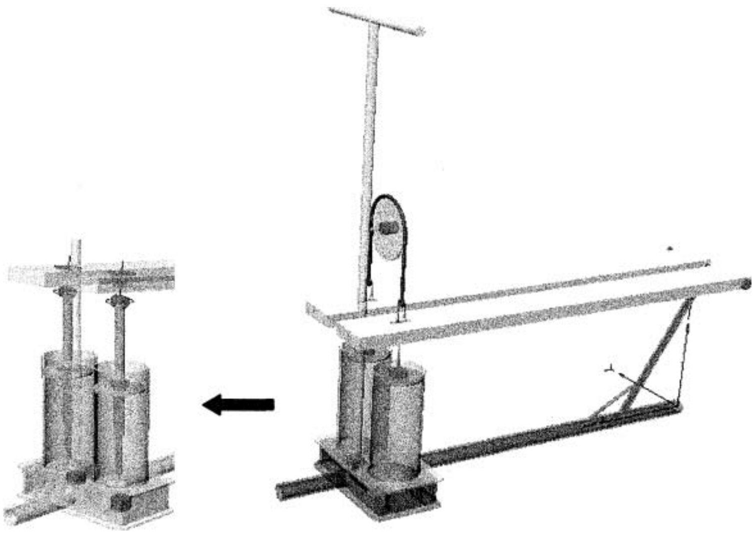


Fig. 41. Treadle pump modeled using MSC.visualNastran software

5 Conclusions

Improving the performance of the treadle pump, taking into account dimensional and structural requirements, manufacturing and maintenance aspects, cost reduction as well as ergonomic aspects is the aim of the present work. A human centered approach has been developed searching for improved comfort and thus optimized user's performance. The main parameters were identified and their limit values defined. Comfortable pumping position requires feet angular positions between -10° and +10°. Treadles must be large enough to be suitable for different operator's height and to allow selection of the optimum position. Additionally, relatively large treadles allow flexibility for the angular position of feet.

Authors have proposed the development of numerical and experimental procedures to study the human body movement in a treadle pump. A biomechanical model in association with a treadle pump parametric model can be used to study the movement of the mass centre during each cycle and calculate the joint reactions and the resulting forces at the user's feet. As a result, improvements in the global performance of the treadle pump can be achieved.

References

1. Daka A.: Development of a Technological Package for Sustainable Use of Dambos by Small-Scale Farmers, PhD Thesis, University of Pretoria, South Africa. (2001)
2. Molden D, Amarasinghe U, Hussain I.: Water for Rural Development. International Water Management Institute. (2001).
3. Kay M, Brabben T.: Food and Agriculture Organization of the United Nations. Treadle pumps for irrigation in Africa, synthesis report. Rome. (2000).
4. Thomas, T.H.: The Performance Testing of Treadle Pumps. Working Paper no. 39, DTU, Department of Engineering, University of Warwick, Coventry, UK. (1993).
5. Srinivas S, Jalajakshi C.: Alternatives to Micro-Irrigation: Evaluation of the Treadle Pump. Economic and Political Weekly, September 18, (2004) 4271-4275.
6. Chigerwe J, Manjengwa N, van der Zaag P, Zhakata W, Rockström J.: Low head drip irrigation kits and treadle pumps for smallholder farmers in Zimbabwe: a technical evaluation based on laboratory tests. Physics and Chemistry of the Earth, 29: (2004) 1049-1059.
7. Perry E.: Low-cost irrigation technologies for food security in sub-Saharan Africa. In: Proceedings of the Workshop "Irrigation Technology Transfer in Support of Food Security", Harare, Zimbabwe, 14-17 April. (1997).
8. DTU – Development Technology Unit, The Treadle Pump, Working Paper 34, Dept of Engineering, Univ. of Warwick, Coventry, UK, (1991).
9. Panero, J., Zelnik, M.: Dimensionamento humano para espaços interiores, Ed. Gustavo Gili, S.A., Barcelona, (2002).
10. Silva, M.P.T., Ambrósio, J.A.C., Pereira, M.S.: Biomechanical Model with Joint Resistance for Impact Simulation. Multiboby System Dynamics. Vol.1 (1997) 65-84.
11. Cappozzo, A., Croce, U.D., Leardini, A., Chiari, L.: Human movement analysis using stereophotogrammetry - Part 1: theoretical background. Gait & Posture. Vol 21. (2004) 186-196.
12. Theobalt, C. Carranza, J., Magnor, M. A., Seidel, H.-P.: Combining 3D flow fields with silhouette-based human motion capture for immersive video. Graphical Models. Vol.66 (2004) 333–351.
13. Ayoub, M.M.: A 2-D Simulation Model for Lifting Activities. Computers ind. Engng. Vol.35, (1998), 619-622.
14. Hostens, I. Ramon, H.: Descriptive analysis of combine cabin vibrations and their effect on the human body. Journal of Sound and Vibration. Vol.266 (2003) 453–464.
15. Wu, G.: Age-related differences in body segmental movement during perturbed stance in humans. Clinical Biomechanics. Vol.13 (1998) 300-307.
16. Aggarwal, J. K., Cai, Q.: Human Motion Analysis: A Review. Computer Vision and Image Understanding. Vol.7. (1999) 428–440.
17. Luo, Y., Wu, T.-, Hwang,J.-N.: Object-based analysis and interpretation of human motion in sports video sequences by dynamic bayesian networks. Computer Vision and Image Understanding. Vol.92. (2003) 196–216.
18. Pers, J., Bon, M., Kova, S., Sibila, M., Dezman, B.: Observation and analysis of large-scale human motion. Human Movement Science. Vol.21. (2002) 295–311.
19. Ning, H., Tan, T., Wang, L., Hu, W.: Kinematics-based tracking of human walking in monocular video sequences. Image and Vision Computing. Vol.22. (2004) 429–441.

Continuous Fitness at Home: Designing Exercise Equipment for the Daily Routine

M.C. Gaspar[1], F. Ventura[2], C. Pereira[3], C.M. Santos[4]

[1] Departamento de Engenharia Industrial,
Escola Superior de Tecnologia, Instituto Politécnico de Castelo Branco,
Av. do Empresário, 6000-767 Castelo Branco, Portugal.
calvete@est.ipcb.pt
[2] Departamento de Engenharia Mecânica,
Faculdade de Ciências e Tecnologia da Universidade de Coimbra,
Pinhal de Marrocos, 3030-201 Coimbra, Portugal
fernando.ventura@dem.uc.pt
[3] Departamento de Engenharia Mecânica,
Instituto Superior de Engenharia de Coimbra,
Rua Pedro Nunes, 3030 - 199 Coimbra, Portugal.
candida@mail.isec.pt
[4] Escola Superior de Tecnologia da Saúde de Coimbra,
Rua 5 de Outubro, 3046-854 Coimbra, Portugal l.
kaleu@sapo.pt

Abstract. The main goal of this work is to design tailored fitness devices that allow performing simultaneously home routine tasks and physical exercise, increasing the amount of physical activity in the daily routine. Physical therapists and design engineers have studied the most usual chores, as well as different types of training, to develop innovative and versatile solutions.

1 Introduction

The most recent recommendations indicate that people of all ages should include a minimum of half an hour of physical activity of moderate intensity on most, if not all, days of the week. It is also acknowledged that for the majority of people, health improvement can be obtained by engaging in physical activity of more vigorous intensity or of longer duration. Physical activity has numerous beneficial physiologic effects, e.g. on the cardiovascular and musculoskeletal systems, but benefits on the metabolic, endocrine, and immune systems performance are also considerable.

To maximize the benefits and results arising from workout it is necessary to perform a moderate amount of exercise on a regular basis. Exercising at home provides a convenient form of physical activity with the major advantage of maximizing time and effort. Nonetheless, conventional home gym fitness devices have several disadvantages such as the space required, their cost and the effective time spent when using them. On the other hand, they are rarely used by most persons, mainly due to the lack of motivation and the need to quit home routine.

Please use the following format when citing this chapter:

Gaspar, M. C., Ventura, F., Pereira, C., Santos, C. M., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 147–160.

2 Current Urban Culture

In current urban lifestyle, daily life common physical tasks have been gradually replaced by machines and appliances. Yet, it is well known that one of the easiest ways of getting adequate exercise is to forego the use of an elevator or other mechanical conveniences and choose instead doing these activities manually, i.e. by one's own body. This type of exercise is fairly easy to do since it is purposeful and uncontrived.

2.1 The Home Routine

Not a long time ago, daily life was made up of a series of physical tasks. Stairs needed to be climbed, dishes and clothes needed to be washed by hand and a series of simple chores were done manually. Although modern progress has increasingly substituted physical activity by automated or mechanized devices, it is still possible to regain the home field to machines by doing manual chores during the daily schedule. As a consequence, performing manually useful household tasks can lead to a better body condition.

Physical exercise is part of one's life and refers to the walking, moving, and lifting inherent in just moving through one's daily schedule. In this work, household and leisure activities at home have been analysed on the pursuit of identifying significant time intervals that can be converted into physical exercise occasions. Figure 1 shows some of the most significant regular activities at home that take more than just simple instants to be performed. Although these activities are usually performed in a steady location with a somewhat rigid posture, they can easily be converted into a physical activity occasion if complemented by the use of proper fitness devices.

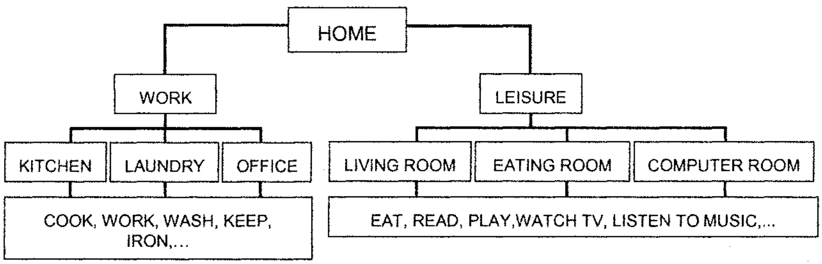


Fig. 1. Ordinary home routine repetitive tasks.

2.2 Personal Health Care and the Need to Exercise

In developed countries, due to sedentary lifestyle and labor-saving devices, many people do not get enough physical exercise. Few of today's occupations require vigorous physical activity and much of the leisure time is spent in stationary occupations. Thus, sedentary and obesity are considered two of the main modern plagues resulting from the lack of physical activity in the current urban habits. Although most of the population is concerned about its physical condition to improve health, looks or athletic performance, in practice they usually do not walk up the stairs instead of taking the elevator.

Because many people in developed countries also suffer from increased job and lifestyle stress, the lack of adequate exercise is doubly harmful. As a consequence, physical exercise is being recommended by almost all medical specialties and the benefits get well beyond the recovering of the musculoskeletal systems. Cholesterol lowering, vascular tonicity, cardiac muscle strengthening, enhancement of aerobic capacity and even beneficial physiologic effects are amongst the major benefits of physical exercise.

There are two types of exercise a person should do in order to stay healthy: aerobic exercise and resistance training. While aerobic exercise means doing something fast and repetitive enough to get one's heart beating 80% faster than normal, resistance training refers to lifting or pushing against something heavy or difficult to move. To obtain the best benefits from aerobic exercise, it should be performed in at least 20 minutes intervals. As such, when trying to adapt home routine into physical exercise, the selected tasks should take similar time intervals. However, considering that more than one household task can be transformed in an opportunity to do exercise, the cumulative time of different chores can have the same type of physical benefits.

3 Conventional Home Fitness Devices

To maximize the benefits and results from workout it is necessary to perform a moderate amount of exercise on a regular basis. Modern urban population spends most of its time between professional activity and home. The increasing work schedules and the unavoidable household tasks lead to a surprisingly short period of effective free time. Exercising at home provides a convenient form of physical activity with the major advantage of maximizing time and effort.

3.1 Commercially Available Devices

Current lifestyle does not permit that most people may walk to work or have a physically intense professional occupation. In such cases, it is necessary to set time aside to deliberately exercise. The need to exercise and the lack of free time turn into a dilemma, which people often solve getting a regular exercise program in any fitness academy. One alternative solution is the use of home fitness devices.

State-of-the-art home fitness devices with the objective of performing physical exercise indoor have several advantages, such as allowing exercise in a regular fashion, at any time of the day or night, and providing a safe and adequate home exercise for all kinds of people. There is a wide range of commercially available equipment devoted to exercise all major muscular groups. Most widely appreciated are steppers, cycles or elliptical trainers [Fig. 2].

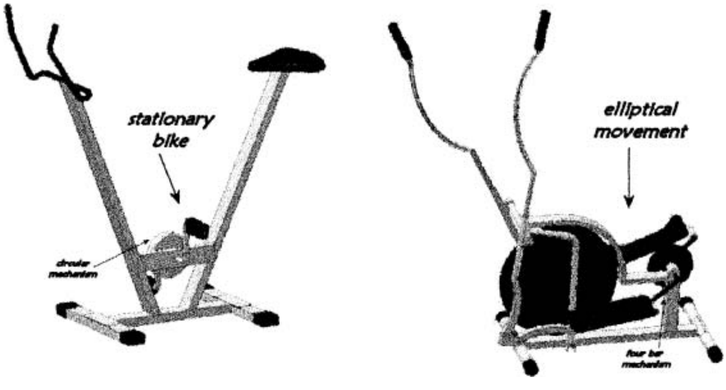


Fig. 2. Conventional home fitness devices.

Home gym equipment might also be a possible solution. Designed for the home, these gyms try to offer a wide variety of full range-of-motion exercise movement in a single group of devices. However, the large space required and the high costs associated are many times inhibiting factors for their widespread.

3.2 Main Restrictions to the Use of Conventional Home Fitness Devices

Very few people can enroll an exercise program for more than a few weeks. Most people are impatient and want to get more done in less time and with less effort. As such, if people are not well motivated, it is very unlikely that they will eagerly free some time on their personal schedule to quit other more rewarding occupations.

Even when fully motivated, the lack of space that characterizes most current urban constructions turn many times into an impossible quest the use of conventional home fitness devices. People have to face the problem of not only managing some space to perform the exercises [Fig. 3], but it is many times very difficult to find free space to keep the equipments when they are not in use.

Since each type of home fitness device is usually dedicated to exercise a specific muscles group, the need to exercise completely the whole body would turn the space problem much more evident. Home gyms try to offer a complete solution, but as referred, the space and cost associated are severe limitations on their use.

As a conclusion, the use of lightweight, compact and easy to move solutions would be a great improvement on the enhancement of home fitness devices.

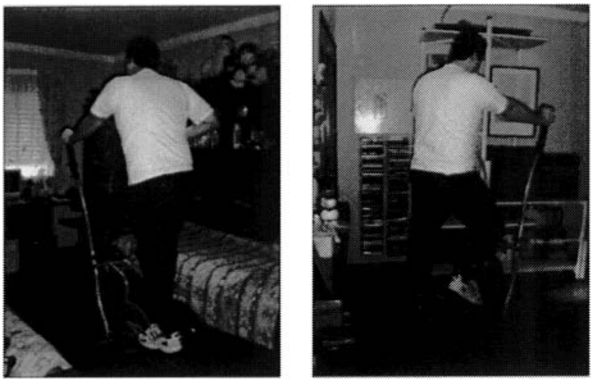


Fig. 3. Performing exercises on a home fitness device.

4 Home Exercise Equipment for the Daily Routine

It is well established that physical exercise is an effective need for current urban culture. However, the difficulties that arise every time a less motivated individual intends to carry out a regular exercise program show that this problem is far from being solved.

This work proposes a commitment between the basic need to undertake regular physical exercise and the lack of time that in general characterize a professional and social active person.

4.1 Main Requirements

As referred before, even the busiest person must perform at least a minimum home routine in order to keep order in personal life. Considering that an individual manages to motivate himself to carry out a physical exercise program at home, there are a few requirements that he tries to fulfill. As a consequence, the more demands he can fulfill, the higher is the probability that he will keep up the program.

In what concerns physical exercise it was previously shown that self motivation is one of the main issues. As such, fitness equipments must be fun to use and, although high technology is an appellative attribute when choosing between similar equipments, the truth is that they must be simple to use. On the other hand, considering that fitness equipments are to be used at home, they must be both easy to move and must occupy the less space possible. Figure 4 synthesizes the three main requirements that the potential user of a fitness device must fulfill.

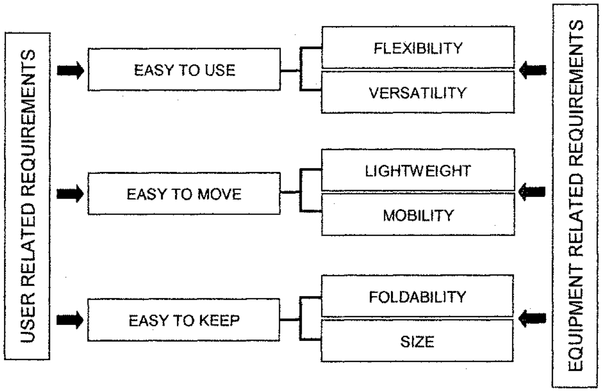


Fig. 4. Home fitness equipment main requirements.

In order to comply with the user's demands, fitness devices must have technical specifications that lead to the satisfaction of the user's preferences and needs. Thus, to be easy to use by any individual, they must present great flexibility to adjust to different persons in a simple and efficient expedite way. They must also be very versatile, allowing their use in many different ways and environments.

The lack of space characterizing the majority of urban buildings makes the ability to be moved and kept in a minimal space two of the major requirements of current fitness equipments. Therefore small equipments are preferred in comparison to larger ones. However, in the latter case, this drawback is minimized by the ability to fold the equipment, which proves to be a good solution. In addition, the weight and mobility of the equipments are intimately related to the need of moving them around the home, between the storing and exercising places.

4.2 Home Tasks *vs.* Fitness Devices

As previously referred, the most common home routine tasks are usually associated to work or leisure. It was also shown that most common work tasks are mainly performed at the kitchen, laundry room or in the home office. When analyzed in more detail, it can be observed that some conventional gait and posture are usually associated to specific home work tasks, like food preparation or serving. It can also be observed that most of these tasks need solely the motion of the upper body, maintaining the lower body completely steady. The movement of the lower body during repetitive tasks would be beneficial for three main reasons: i) reducing the possibility of deficient blood circulation; ii) avoidance of long steady postures and iii) the movement associated to physical and psychological benefits.

Figure 5 shows some of the most usual home routine repetitive work tasks as well as some possible fitness solutions in order to perform simultaneously physical exercise. These tasks were analyzed considering the ergonomic guidelines proposed by Landau *et al.* [1,2], while design solutions took into consideration the recommendations by Kroemer[3], Panero *et al.*[4] and the Geneva International Labour Office[5].

Fig. 5. Home routine work repetitive tasks.

In Figure 6 some of the most usual long leisure situations at home are illustrated. As well as for the home work tasks, some possible fitness solutions are proposed, allowing leisure activities and physical exercise in simultaneous.

Fig. 6. Home routine leisure tasks.

4.3 Resulting Basic Mechanisms

When considering the design of physical devices from the mechanical point of view, some simple considerations must be taken [6]: the mechanism must be decomposed into functionally independent systems allowing the subsequent generation of concepts.

Observing the different home routine situations of Figures 5 and 6 it is possible to synthesize the main mechanisms involved in the various proposed fitness solutions. Figure 7 summarizes the basic mechanisms presented in Figs. 5 and 6 and their resulting combinations.

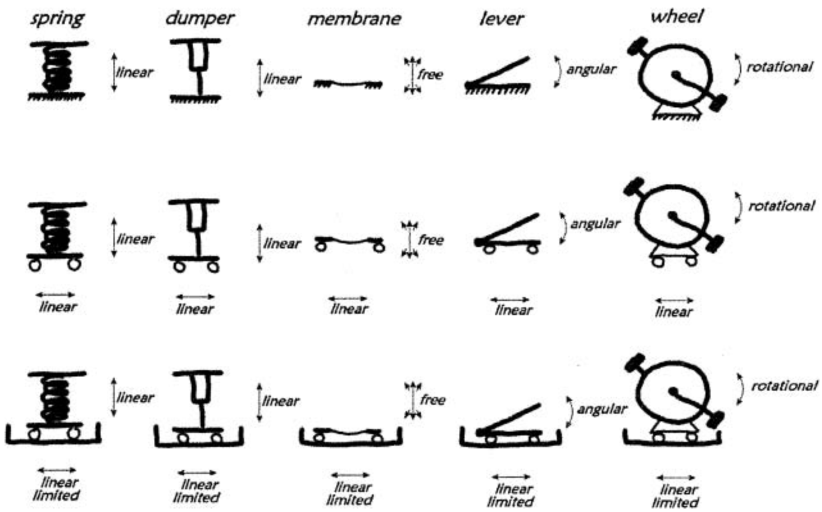


Fig. 7. Basic mechanisms of several home routine fitness solutions.
Home routine fitness solutions base mechanisms.

5 Designing New Solutions

The conceptual design phase is one of the most rewarding; yet, it is not free from failure. Considering that one of the major goals of this phase is to spend the least amount of resources in deciding which concepts have the highest potential to solve the desired application problem, attention will be mainly focused on the assumptions presented in previous sections.

Design guidelines [6-8] were considered to enhance the efficiency of the proposed solutions. The concepts presented in the next sections aimed to fulfill the maximum requirements presented before in order to maximize the features in each solution.

5.1 Compact Assembling Solution

From Figures 5 to 7 we can conclude that all solutions, with exception of Figs. 5d) and 5h), are small-sized, being usually lower than 40 cm height. As a consequence, it can be anticipated that one of the possible concepts focuses on a compact solution that may be easy to carry and to store. Additionally, since versatility is another major requirement, then the ability to have a base structure where different types of exercises can be performed is also very appellative to the potential user.

A linear roller is proposed in the sketch of Figure 8 based on the solutions proposed for the work task of Fig. 5f) and the leisure situation of Fig. 6). The base structure permits not only that the roller gets aligned with the user's foot/leg, but also that it does not roll away from the user. The rolling "gap" may be easily adjusted through simple fasteners, as illustrated in Figure 8. This solution may be used alternatively by each foot, requiring no effort to change the exercised leg. The type of exercise herein proposed may foresee that exercising periods of 10/15 minutes are perfectly acceptable.

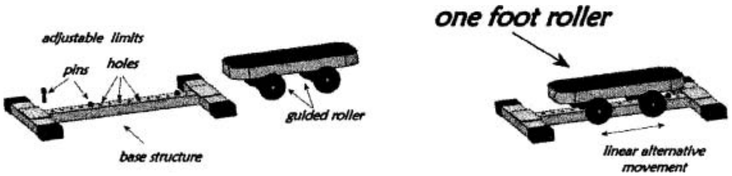


Fig. 8. Sketch representation of a one foot roller exercise device.

In the sketch of Figure 9 a conventional dump-based stepper is proposed in association with a linear roller. This solution is based both on work and leisure tasks, allowing that combined or simple exercises may be performed, which enhances the variety of possible alternative workouts.

The simple grip-components allow a quick and easy fixing of the stepper on the roller, simplifying the intermutability of exercise devices. When the roller is fastened, a pure stepper exercise can be performed, whereas with the roller loose, a combined exercise is possible.

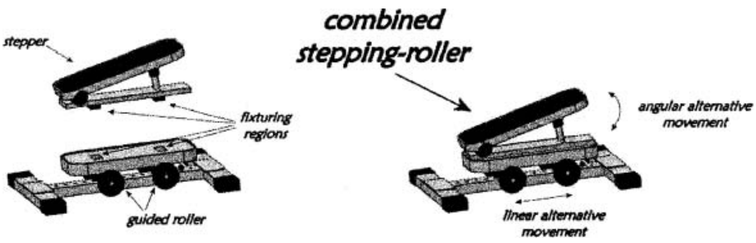


Fig. 9. Sketch representation of a combined stepping-roller.

A compact stationary bike is presented in the sketch of Figure 10 based on the solutions proposed on the leisure situation of Fig. 6d). The base structure permits an easy fixing of the bike pedals. The relative position of the device in the base structure can easily be adjusted to every type and size of the potential users. If necessary, a height adjustment can also be included.

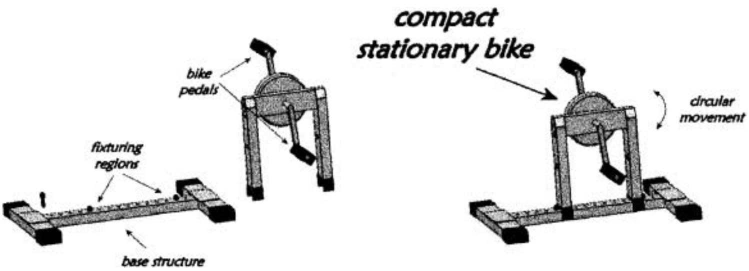


Fig. 10. Sketch representation of a compact stationary bike.

The complete range of devices of the preliminary version for the compact assembling solution can be observed in Figure 11. It includes three main types of exercises and may also allow the inclusion of the remaining mechanisms illustrated in Figure 7. As an example, the inclusion of the membrane stepper can easily be accomplished with the use of a frame to be fitted on the top of the roller. For the horizontal spring and dumper devices, a similar solution can also be considered.

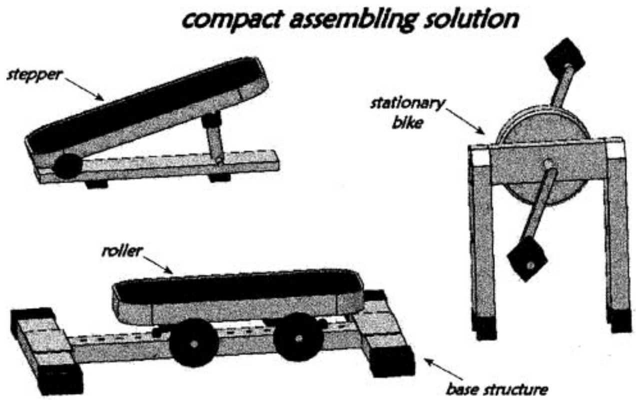


Fig. 11. Home routine exercise devices of the compact assembling solution.

5.2 Versatile Elastic String Jogger

When considering elastic-based joggers, three immediate advantages permit selection of that solution on flexible, lightweight trainers. One of the main advantages of a membrane-type exercise device (or of a long elastic string solution) is related to the increase of effort required to reach larger displacements of the elastic components. Unlike what happens when working out with simple weights, where the strength needed to raise the weights remains constant, on elastic components the resistant strength increases with the applied stroke.

The second advantage of the elastic-based joggers is the versatility of movements allowed, imposing no directional restraints, and thus, easier to be used. It can also be easily adapted to almost every type and size of users without major adjustments.

The third, but not the least advantage is related to the ability of working out not only the lower body, but also the upper part of the body. As seen in Figures 5d) and 5h), the simple home routine tasks can be performed with a small effort needed to oppose to the resistant elastic string, when working out not only the lower members, but also the upper members of the body. To improve the ease of use, fast wrist and ankle fixing devices allow a simple and expedite assembling and loosening. Figure 12 illustrates these solutions.

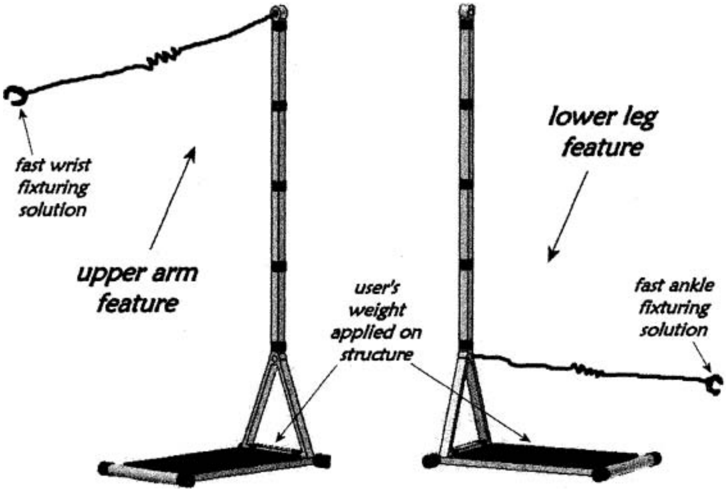


Fig. 12. Upper and lower body exercise solutions on an elastic string jogger.

The complete range of features of the elastic string jogger proposed in this work can be observed in Figure 13. It includes the folding or clamping regions that permit an easy disassembling of the device, as well as the folding joints to pivot the triangular frame. Although it is intended that this device should be very resistant and with a low weight, its ground stabilization is guaranteed by the user's weight. Additional features can also be included, like the anti-slip rubber coating, enhancing the security of use associated to this innovative device."

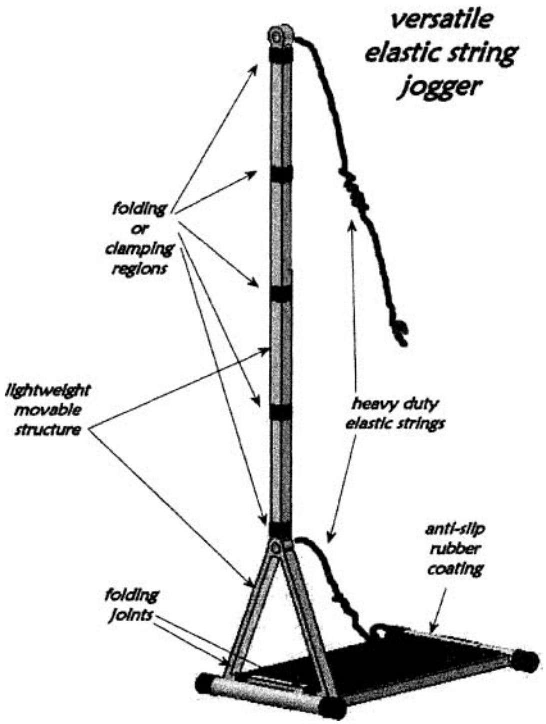


Fig. 13. Sketch view of the versatile elastic string jogger.

6 Preliminary Conclusions

The present work points out several solutions to address the problem of motivating people to do exercise regularly and, thus, become healthier. The most usual home routine tasks were studied and different forms of training were considered in order to obtain conceptual designs of innovative fitness solutions. As a result, tailored fitness devices are proposed to allow performing simultaneously home routine tasks and physical exercises. Two preliminary solutions are presented. A compact assembling solution including combined features characteristic of a roller, a stepper and a stationary bike is presented. A versatile elastic string jogger is also presented, allowing physical workout of the user's upper and lower body regions while performing regular home tasks.

References

1. Landau, K., Rohmert, W., Brauchler, R.: Task Analysis: Part I. In: Mital, A., Kilbom, A, Kumar, S. (ed.): Ergonomics Guidelines and Problem Solving, Vol. I. Elsevier Science (2000)
2. Brauchler, R., Landau, K.: Task Analysis: Part II. In: Mital, A., Kilbom, A, Kumar, S. (ed.): Ergonomics Guidelines and Problem Solving, Vol. I. Elsevier Science (2000)
3. Kroemer, K.H.E.: Ergonomics: How to Design for Ease and Efficiency. 2nd edn. Prentice-Hall (2001)
4. Panero, J., Zelnik, M.: Human Dimension & Interior Space. Watson Guptil Publications (2002)
5.: I.L.O.: Ergonomic Checkpoints. Geneva International Labour Office (1996)
6. Ullman, D.G.: The Mechanical Design Space. McGraw-Hill (1992)
7. Shigley, J.E., Mishke, C.R.: Mechanical Engineering Design. 5th edn. McGraw-Hill (1989)
8. Norton, R.L.: Machine Design. An Integrated Approach. Prentice_Hall (1996)

Design Sketching for Space and Time

Simone Rozzi, William Wong, Paola Amaldi, Peter Woodward, and Bob Fields

Middlesex University
Hendon London NW4 4BT
United Kingdom

s.rozzi, w.wong, p.amaldi-trillo, p.woodward,
b.fields@mdx.ac.uk

Abstract. In this paper we present a case study of how design sketching can be used as a technique for exploring and creating a common understanding between users, designers and software developers, of the representation design requirements for supporting spatial-temporal reasoning in Air Traffic Control (ATC). The safe and expeditious control of aircraft requires the ATC controller to think in terms of 3D air space, and also plan ahead in time. We refer to this mental process as spatial-temporal reasoning. ATC is a 4D (3D plus time) problem but is currently supported by 2D tools such as the Plan Position Indicator-type radar displays that are seen in ATC centres. This requires the air traffic controllers to construct mental models of the air traffic situation to ensure safe vertical and horizontal separations between moving aircraft, and also expedite traffic flow. These objectives require prediction of traffic patterns and potential bottlenecks. To explain how we used design sketching, we report on the Task Analysis of an exemplar ATC task, and the characterisation of this task in spatial-temporal terms, and how the Ecological Interface Design principle of visualisation of constraints was applied to guide the development of the 4D visual form of the representation design.

Introduction

The purpose of this paper is to present a case study which was used to explore how design sketching could be used to elicit and understand the spatial-temporal issues in designing interfaces for supporting space and time applications, such as air traffic control. We will present a preliminary framework for discussing the interaction between spatial and temporal aspects of work and their effect on representation design, as well as discuss how design sketching, informed by the framework, can be used to facilitate the creation of a common understanding of the spatial-temporal issues of the problem and the design space, between users, analysts, designers and software developers. This work is part of a European Commission funded project aimed at the development of a 4D Human Machine Interface for Air Traffic Control (ATC).

The role of ATC is to ensure the safe and expeditious movement of aircraft within the airspace under its control. This requires the ATC controller to think of aircraft movements in terms of 3D space, de-conflicting aircraft travelling at different altitudes, speeds, and in various directions, to intercept a point in space, to

---

Please use the following format when citing this chapter:

Rozzi, S., Wong, W., Amaldi, P., Woodward, P., Fields, B., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 161–183.

join or sequence aircraft into a queue, and then to predict the likely future situation and then to plan ahead in time. We refer to this mental process as spatial-temporal reasoning. The ATC controller is currently supported by 2D tools such as the Plan Position Indicator-type radar display which show only the plan view of aircraft, showing only horizontal separations. Aircraft altitude information is provided as a text label tagged to each aircraft's radar blip, requiring the ATC controller to create a mental model of the 3D space in order to assess vertical separations. Planning and de-confliction is supported by tools such as the arrival and departure manager software and short term conflict detection software. Aircraft unfortunately do not only fly straight and level, but also climb and descend, at different rates of climb, and often turning at the same time. In addition, the controller also has to project the situation into the future, based on current and past aircraft track behaviours, as well as instructions that are likely but have yet to be given. This is the temporal or what some refer to as the 4^{th} dimension, when referring to a 3D + time, or what we now refer to as a 4D display. The development of such a 3D mental picture [1] of the air traffic situation and then projecting it over time, 4D, represents a significant challenge for the expertise of air traffic controllers, often taking several years of training and experience to develop. Our project team was presented with the challenge of developing a visual format that could support such 4D visualisation and interaction needs.

To meet this challenge, we conducted a task analysis to first understand the spatial-temporal nature and cognitive demands of the ATC task. This has been briefly outlined above and will be elaborated further on in this paper. The task included a cognitive task analysis using the Critical Decision Method in-depth interview to understand the reasons for their actions and considerations; field observations including several hours of video recordings in order to identify the patterns and structure of work and information handling behaviours within the context of the work environment; and Contextual Inquiry interviews which were done in-situ near the ATC controllers' workstations to understand their actions.

Design sketching is a rough and quick drawing technique where the intention is to capture, represent and illustrate key features of the problem that need to be visualised. We used the rough sketches to articulate, visualise, communicate and clarify our understanding of the spatial-temporal requirements and hence the nature of the likely visualisation support needed during discussions with users, other analysts, designers and software developers. Because the designs were sketches, we could very quickly clarify, and refine concepts and ideas by quickly adding to or taking away from the sketch. We also applied the technique of storyboarding to step through temporal issues such as how to show future states of an air traffic situation in 3D space over a period of time. The Ecological Interface Design approach has a number of design principles, of which the visualisation of constraints is one. We used this principle to focus our attention on developing time into spatial representation terms.

Design Sketching

A design sketch "... should be simple, starting with rectangles, names, and simple descriptions of relationships between functional areas ..." [2]. Lauesen has developed a systematic approach that supports and expands this approach. He introduced the concept of the virtual window, i.e. a 'user-oriented presentation of persistent data' [3]. which can be synthesized as a framework to move systematically, from information requirements to the graphical representation of information. This should occur along three phases: (1) the definition and grouping of information according to the relevant user tasks to be supported; (2) the planning of the data to be presented in each virtual windows; (3) the design of the interactive features such as button, menus, and icons. Lauesen does not mention directly the use of sketches, but the virtual window provides a form of sketching path along which the appearance of a set of interfaces can be sketched and developed. Finally, a last group of contributions explain the importance of using low fidelity paper prototypes at the beginning of the design process. These low quality, sketchy interfaces, sometimes using sticky Post It notes and paper cut-outs of GUI elements such as windows and drop-boxes, enable the design team to evaluate the working of the concepts, and in the process identify early usability issues while the interface is still being designed [4].

The literature on software user interface and interaction design, and in particular, sketching as an early design technique, is somewhat limited. The literature in the requirements engineering domain usually emphasises the description, organization, negotiation and verification of the requirements along the interface development process. Alternatively, we find different design principles based on Human Visual Perception [5] and Human Factors [6] principles that support good display design practices. These contributions create the context of action for the system designer, who, based on the requirements and assisted by good design principles will define the interface appearance. This allows them to move from early interface designs to the definition of the final layout. However, there has generally been only a few contributions which explain the role of sketching in this process.

One such example is from Buxton [7] who suggests that design sketching is fundamental in the early ideation phases, but "...due to the temporal nature of what we are designing ..." i.e. the need to consider behaviour of both users and system interacting over a period of time, "... conventional sketching is not adequate" [7]. In other words, design sketching for user interface and representation design, does not have the maturity that the same technique has in fields such as industrial design or architecture. Design sketching techniques for software user interface development need to be further developed to cater, for example, for the dynamics of the spatial-temporal nature of human work demands. Some of these demands include the avoidance of biases in temporal decision making, minimising temporal errors, and maximising temporal awareness [8].

Air Traffic Control and the Approach Sector

Air traffic control is a service provided for the purpose of preventing collisions between aircraft and expediting and maintaining the orderly flow of air traffic. The type of air traffic control services may be divided according to the different aircraft control phases. As an aircraft take-off is handed over from the controller in the airport Tower, to the controllers at the Area Control Centre or ACC. The controllers at the ACC use radar to guide, sequence and de-conflict arriving, departing and transiting aircraft, as they fly, climb or descend, through this airspace known as the Terminal Manoeuvring Area or TMA. This area is sometimes referred to as the Approach Sector, and the controller, the Approach Controller. Once the aircraft clears the TMA, the aircraft is handed over to the En Route Controllers who will manage the traffic, usually at a higher altitude, between the respective TMAs.

The de-conflicting activity of the controllers is a very cognitively demanding task. It requires the controller to maintain a dynamic mental picture of several aircraft climbing and descending within a relatively limited portion of airspace, and over a short period of time. At one point during our study, we had observed an Approach Controller sequencing 12 aircraft to land at an airport within 10 minutes. For each of these the approach controller must ensure the safety sequencing and separations, i.e. aircraft must be under no circumstances closer than 3 miles horizontally and 1000 feet vertically.

In the approach control three controllers usually operate together. A coordinator takes aircraft from the en route sector. S/he makes a plan about 10-20 minutes in advance about aircraft trajectories towards the airport. Then as the aircraft gets closer to the airport they are sequenced and separated by the approach controllers. These working positions can include one to three controllers depending on traffic conditions.

Rationale for Introducing 4D Technology into ATC

Approach Control represents one of the domains of interest for the AD4 project. Due to the lack of important 3D aspects of information such as altitude and 4D aspects such as projected future arrival sequences, 2D PPI radar displays are often augmented with these additional information, e.g. altitudes tagged as text labels, and arrival sequences packaged within decision support tools such as the Arrival Manager software.

Ironically, while 3D and stereoscopic displays can contribute towards an approach to resolving the information overload problems of 2D radar representations, it also poses new challenges for the display design. Recent work suggests that the pictorial realism often found in 3D user interfaces does not necessarily improve operator work efficiency. Instead, it can lead to breakdowns in performance that increase visual complexity, perceptual error, and mental effort [9]. Therefore, part of the AD4 project efforts have been allocated to the investigation of the type of representation form that could effectively support the controller's spatial-temporal reasoning in the approach control, without unnecessary photo-realism.

Methodology

The study investigated the control task performed by Approach Controllers at the Rome Approach Control Centre, over two blocks of three days each. The study combined a number of techniques. We conducted a number of cognitive task analysis interviews using an approach similar to the Critical Decision Method, investigating difficult or non-routine situations in order to identify the strategies they invoked and the cues they attended to when sequencing and de-conflicting aircraft in 3D space and over time. We also observed and video recorded how controllers operated under routine conditions, paying special attention to patterns of behaviours and information handling practices involved in coordinating actions between controllers and planning ahead. In between these observations as the opportunities arose due lower workload, we used Contextual Inquiry techniques to interview the controllers in the context of their natural settings. This allowed us to ask the controllers to explain their rationale for a decision and they can very quickly walk-through their decision while the aircraft are still on the radar display, as is the case in the example being reported.

The results were further analysed within the context of the Ecological Interface Design principle of making constraints visible, to derive an understanding of what might be needed to represent space and time as visible constraints. This approach implies the direct visualization of the cognitive workspace and the associated constraints for the operator. For example, this involves computation of distances, altitude, speed and rate of climb at a particular distance. These cognitively demanding computations can be allocated to the display itself by making visible these parameters in relation to the aircraft and its flight. These concepts were encapsulated into the specification of scenario based user requirements (i.e. what the user needs to know), as well as visual requirements (i.e. what the user needs to see).

Task Analysis: What Are the Spatial-Temporal Demands of the Approach Controller Task?

This section describes one of the many control situations studied during our investigations. It is useful as an exemplar of the cognitive complexity typically encountered in ATC work. It describes how a controller evaluated vertical separations and subsequently de-conflicted aircraft trajectories projected forward in time. The speech bubbles in the diagrams have been translated from the Italian. Using this scenario and other scenarios studied but not reported in this paper, we derived a series of 4D representation design needs. The next section will then report on the use of design sketches to translate those spatial-temporal needs into early representation forms of the user interface.

6.1 The Scenario

Flight LOT 314 (see Figure 1) had just taken off and is now under the control of the Approach Controller. It is currently at FL (Flight Level) 140 (or 14,000 feet), and will climb north-eastwards to an en route altitude of FL 260. At this point, two climbing paths are possible: (i) the standard one (dotted line, Fig. 1), or (ii) a short cut (continuous line, Fig. 1). The short cut would allow saving both aircraft fuel and time necessary to vacate the sector. In other words it would improve air traffic efficiency. However before being implemented a several considerations are needed.

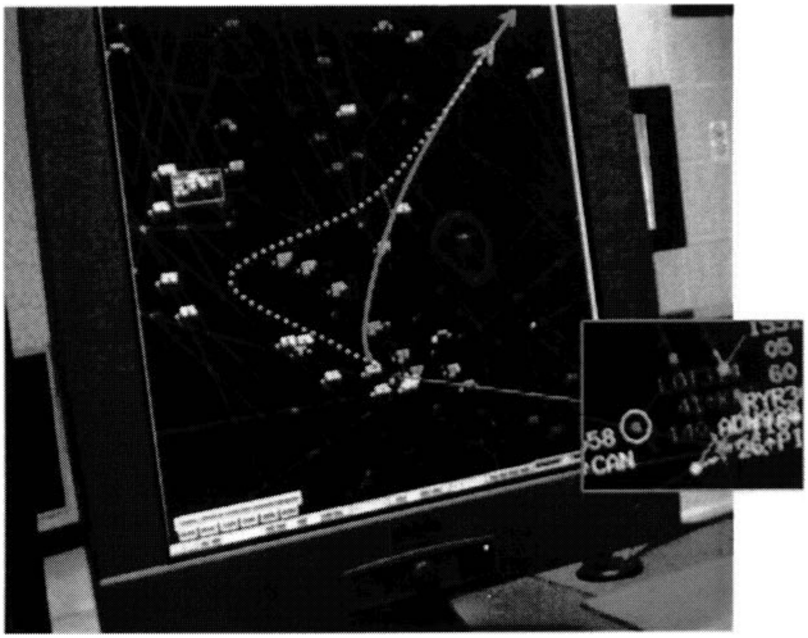


Fig. 42. A view of the radar used for approach control. An aircraft, LOT314 has just taken off and the controller must decide whether to allow the aircraft along the standard climbing departure route (dotted line) or along a short cut (continue line)

The controller notices that two groups of aircraft from the north and North West will cross the "short cut" trajectory of LOT314 (see Fig. 2). He will therefore need to estimate the existence of safe separations in the immediate future, i.e. will the crossing traffic be closer than 3 nautical miles horizontally and 1000 feet vertically.

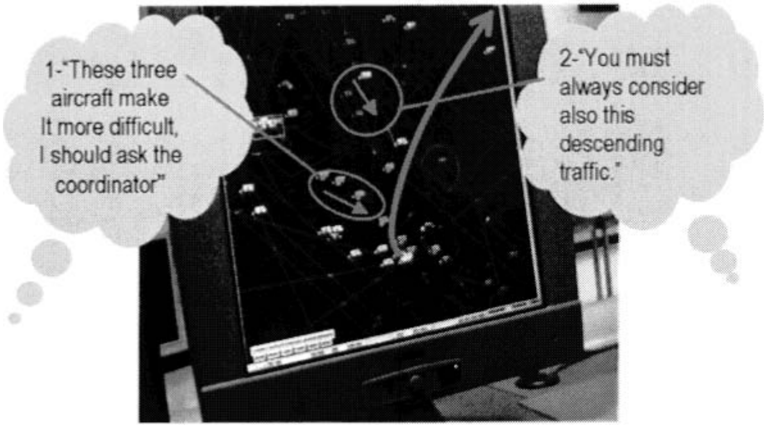


Fig. 43. Traffic patterns (circled in the picture) considered by the controller before deciding whether it is possible to allow the LOT314 to climb along the short cut (the main arrow leading off to the right). The arrows within the ellipses indicate the direction of the two groups of aircraft. The balloons report the controller's explanation

In order to assess if there is an adequate safe separation (see Figure 3), he mentally projects the position where LOT 314 will be and the path it will likely to take in the next two to three minutes. He then checks which other aircraft is going to be near the projected position or path.

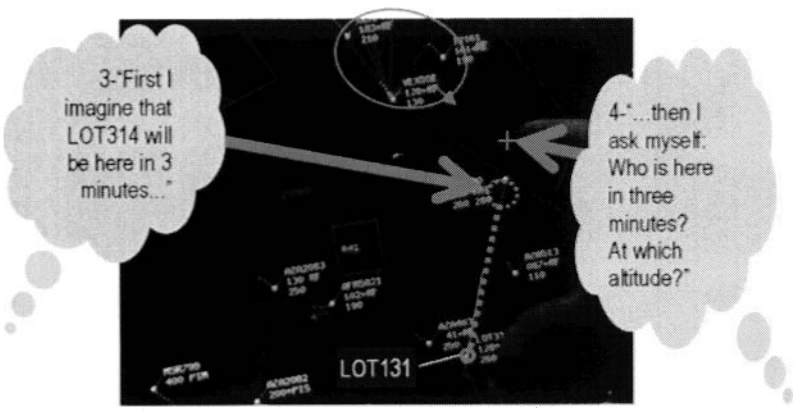


Fig. 44. Short term assessment of separations. The end of the dotted line represents the expected position of the LOT in three minutes. The controller also assesses which other aircraft is going to be close to this area in the same time frame. For each aircraft likely to intersect with the new shortcut on the horizontal plane an evaluation is made for the vertical plane (Flight Level), this is to ensure that the safe separation limits are kept

The LOT aircraft, as mentioned earlier, is climbing from FL 140 to FL 260. In the first assessment in Figure 4 (Note 5) the controller is considering that Flight

I2161 is descending from FL153 to a lower flight level, and assesses that no conflict exists. In the second assessment (Note 6) the controller notes that ELM 71 is descending as well from FL 170 to a lower one. As the rate of climb of the LOT aircraft is 4000 ft per minute, it is likely to be at FL180 in the intersection point. Given that in both the situations the separations exist the controller can allow the pilot to take the short cut.

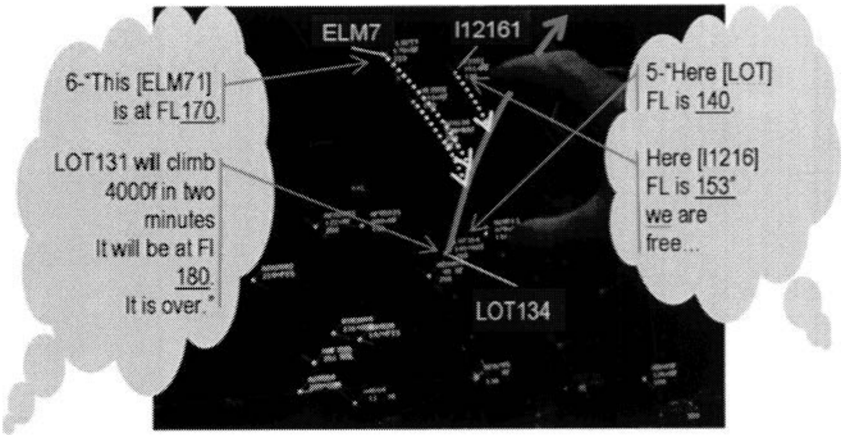


Fig. 45. Controller's assessment of vertical separations. As the controller knows that some aircraft are going to cross the trajectories of the LOT134, he evaluates the existence of safe levels vertical separation (see the text for explanation). The square brackets report the name of the aircraft considered by the controller during the assessment. These were not mentioned by the controller as he was referencing them pointing to the monitor

6.2 Implication for the New System

The Actual Radar Display and Its Limitations. The current radar display provides a top 2D view of the sector being controlled. This makes it very clear for the controller to understand the horizontal displacement of air traffic and the associated separations, actual and in the near future. However, this does not cater fully for the approach controller's needs such as those described in the example scenario where the approaching traffic required a supervision of climbing and descending aircraft, i.e. traffic moves both horizontally and vertically. The actual radar display makes it difficult to understand 3D conflicts (converging laterally and vertically), as vertical information is provided for each aircraft only by a numerical indication of the current flight level. This requires the controller to make effortful mental computations of the likely aircraft flight paths.

Resume of the Controller's Problem: the Projection of Aircraft Vertical Position in Space. As the episode above demonstrates, the controller, using his or her experience of previous traffic patterns, is usually able to make a reasonably accurate projection about the horizontal position that a given aircraft is likely to occupy in the near future. Unfortunately this is not so when determining the aircraft's next vertical position. This requires the following mental calculation to be made for each aircraft that must be de-conflicted:

(1)Rate of climb (or descent) x (2) time

As the rate of climb information (1) is usually available only to pilots and not to the controllers, an explicit request from the controller to the pilot is needed. Therefore, such computations will require a communication between the controller and pilot, limiting what the controller can do during that very short time frame, as s/he would be engaged in a radio communication with the pilot. Thus, to ascertain the aircraft's vertical position, the controller needs to perform a mental computation, and also a communication with as many pilot as many the number of aircraft that must be de-conflicted.

Based on the cognitive task analysis of this incident and other similar incidents, one of the main contributions a 3D user interface could offer to a controller would be to make visually salient the vertical and angular separations between aircraft in the 3D space. This could potentially reduce the effort which currently requires a demanding mental computation by the controller to make. A number of visual representation requirements resulted from the work analysis, and they include the following:

(a) *Show the intersection point in 3D space, when required by controller.* The 2D radar scope must still available to controller so to avoid losing the pattern recognition skills that controllers have developed with this tool. The 2D scope would serve the identification of critical traffic, these could be then accessed with a 3D on demand, in order to verify the associated vertical separations.

(b) *Show estimated vertical, horizontal and angular separations expected at the predicted intersecting point.* Based on the work analysis it was felt that making visually salient these variables would determine a quicker controller decision making.

(c) *Visualize the rate of climb (or descent) of the aircraft.* In this way it is available to the controller an important data that today is available only to pilot;

(d) *Show the heading of the aircraft and whether the aircraft is navigating on an assigned flight level, or is climbing/descending to a new FL.* This was felt to be fundamental to provide to the controller an indication of the position that the aircraft would be likely to occupy in the near future.

Framework for Spatial Temporal Representation Requirements

The primary focus of our work in this project is to understand what 4D aspects of the air traffic control task need to be represented and how they should be best rendered. As described earlier in this paper, this task is very much spatial and temporal in nature: the controllers need to visualise in their mind the 3D positions of the aircraft in relation to key features of the airspace environment, and in terms of time, as they coordinate current actions and plan ahead. In this section we propose a framework for considering the elements of 3D space and time, and in so doing, identify the goals, functional relationships and entities, that need to be represented in an ATC control interface.

Our cognitive task analysis suggests that the cognitive work that is performed by air traffic controllers can be generally described along two dimensions: spatially and temporally. *Objects* exist in 3D space, and these entities include aircraft, airways, fixes, beacon points, and terrain. Objects can also have attributes, e.g. aircraft labels provide additional information such as aircraft call sign, model, and more dynamic information such as altitude, which defines their position in space. These objects are subject to *constraints* such as the permissible distances between objects such as aircraft. We call these proximities separations and minimum altitudes. When operating in certain localities or over certain terrain, the aircraft may be subject to other constraints, for example, where they can fly over or the minimum safe altitudes they should transit the area.

While these two groups of information are often adequate to develop visual spatial representations of work domains such as ATC and even emergency ambulance control, they are often inadequate for representing the work of the domain so that the controllers can perform their jobs effectively. The objects and constraints can show where the aircraft are and are heading, and the constraints such as separations and distances from each other as well as larger obstacles such as weather. However, controllers also need to know other *relationship*-based information. One such fundamental relationship in activity-based systems, is the *who-what-where* relationship: Who is that aircraft? what is it doing? where is it going? Implicit in this relationship is the aircraft relationship to (i) other aircraft, to determine if their trajectories are likely to cross and if they are likely to come within too close to each other, or (ii) obstacles, i.e. their likelihood to come close to some larger obstacle such as heavy weather or no-fly zone. Together these three sets of information allow us to know what is happening to the activity being controlled. This is illustrated in the Spatial Dimensions half of the Spatial-Temporal Framework in Figure 5.

Here, the spatial parameters of the work domain identified above are static. They represent the state of an aircraft at a point in time. Navigation beacon points and airways boundaries, for example, do not change either. Again, this is therefore inadquate to fully represent the work in a dynamic environment such as ATC. The dynamics of the environment are the interactions between the objects and its constraints, and hence the relationships that emerge from those interactions. We have referred to this as the *behaviours* in Figure 5.

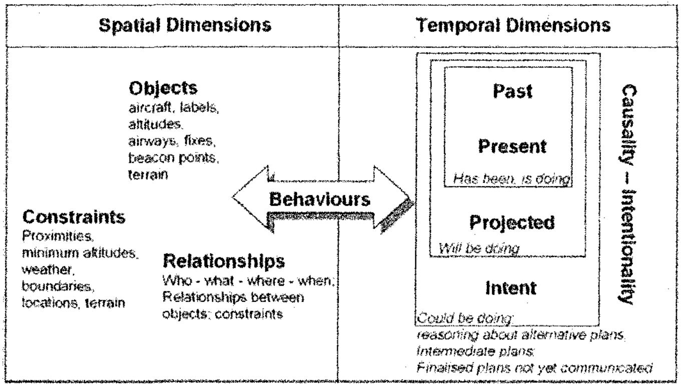


Fig. 46. The Spatial-Temporal Framework

As interactions can only occur over time, behaviours, or sets of interactions, integrate the spatially-based activity with the temporal dimension of the framework. Behaviours or events can be viewed in terms of different time horizons which have been called time bands [10]. Each time band being relevant to different aspects of the event, such as second to second for real time control and coordination activity, the day or the month it happened in terms of reporting and debriefing activity. This notion of time banding allows us to consider how different levels of activities such as planning and coordination occur, and the processes that should be supported.

However, in ATC, longer timeframes such as days and months are not directly relevant to real-time control activities, although air traffic patterns of behaviours over months and years are important for longer term (time band) activities such as infrastructure planning. From the real-time ATC control perspective, the real-time control time band of seconds and minutes, is the time band of focus. Behaviours within this time band may be described in terms of their (immediate) *past*, where they are and what they are doing at the *present*, and then based on this track history, predict what they might be doing in the *projected* immediate future. For instance, an aircraft that has been (past) and still is (present), flying straight and level at 27,000 ft, heading 020 degrees, with a speed of 460 knots, can be projected to be at a point bearing 020 degrees and 38.3 nautical miles away.

Intended behaviours are however, not projectable in the same way based on past or historical data. In the scenario cited earlier in this paper, we described how a controller is projecting into the future the intention to send an aircraft through a short cut. If we were to use the aircraft track history at the time the decision was being considered, i.e. the aircraft was climbing and turning as it departs the runway, the projected position will be erroneous. Instead, the controller was making use of a different, non-existent set of data: *if* the aircraft were to be at a point A and *if* it were to take the short cut, what *would* be the *possible* conflicts with its trajectory? This is not based on 'hard' past performance data from which we can compute effects from a cause and perhaps, draw projected positions on the display; but on 'soft' intentional data that are yet to exist in the control system, and therefore the system cannot automatically compute and render a future position. Such a distinction has been referred to by the terms causal systems and intentional systems [11]. In causal systems, the outcomes are predictable by the laws of nature, whereas, intentional systems cannot be and are instead based on human motivation.

As we use design sketches to study how a future 4D visualisation and interactions would work, we started to observe the need to make distinctions in behaviours over time. Our design sketches need to reflect the wide spectrum of activity that a controller needs to respond to or initiate, that is distributed within space and over time, and with information based on actual data, and also, based on intentions. In other words, following from the Law of Requisite Variety [12], the designs that our sketches explore need to have/possess the capability to support the variety of situations that the task domain is likely to present the controller, otherwise, the misfits can lead to performance degradation.

The following sections illustrate how the identified tasks were translated to show spatial and temporal issues.

Communicating the Spatial Temporal Nature of the User Problem

In order to translate the requirements derived from the work analysis into the visual designs, it was felt that a design dissemination workshop was needed to share the researchers' understanding of the control problem, and to start to visualize the form of the novel 4D display. The design team included about ten people, with a background respectively in Human Factors, Information Visualization, Safety, 3D Programming, and Management. The following subsection explains the role of the sketches during the meeting. All of the sketches here reported were drawn on large 50x72cm sheets of paper.

Initially dissemination of the results of the analysis was intended to be achieved by means of a Power Point presentation to the other team members. However this was inadequate given the complexity of the factors involved in the controller's job. Therefore some first sketches were drawn so as to explain the user problem, especially to team members not non-familiar with the ATC domain.

Sketch in Fig. 6 depicts the phases of flight an incoming aircraft. When approaching the airport usually an aircraft enters the controlled sector (see drawing point d), then it must be directed towards the airport along a series of fixed points (the 'fix', and can be thought as a fixed point in the 3D space represented as a longitude, latitude, and an altitude). The controller in this phase can give instructions concerning aircraft speed, route, flight level, so to take the aircraft to the final approach fix, where the Instrument Landing System begins along the glide path(see drawing point a). In some cases such as emergency situations, aircraft can be placed in a holding stack (see drawing point c) while the controllers attempt to give landing priority to the aircraft with the emergency. This example shows how we used the sketch to discuss with controllers and other designers, the behaviours of the objects and constraints during the landing and sequencing task.

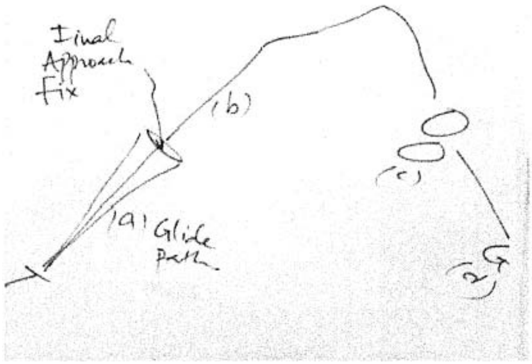


Fig. 47. A sketch of aircraft approach phases

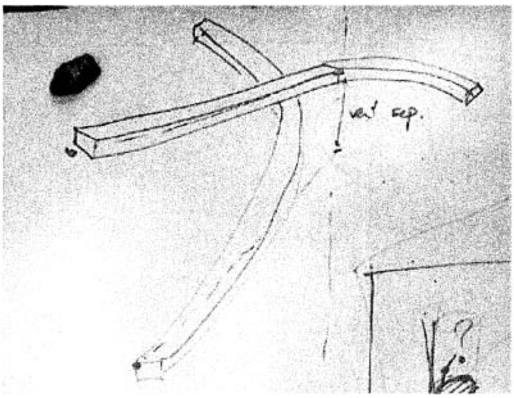


Fig. 48. **Two aircraft climbing paths whose trajectories intersect. In this instance the controller need to evaluate the existence of vertical separation**

Sketch in Fig. 7 depicts two aircraft whose trajectories cross at some point in space and time in the near future. In these situations the controller usually acts beforehand so to ensure that the aircraft will be adequately separated when passing through the intersection point (when the respective aircraft flight path will intersect). The separation is represented at the centre of the drawing and this is what the controller estimates three minutes before the aircraft reaches the corresponding position. The trajectory is sketched as a volume of square sections to articulate the safety boundaries of the aircraft in the 3D space. Aircraft do not just fly in straight lines, but need to operate in a volume of space in order to ensure safety. In addition, aircraft can also turn while climbing and the resulting trajectory is a curve in the 3D space with a turning radius, again, rather than straight lines joining at a sharp angle to each other.

Overall, the sketches in Fig. 6 and 7, up to this point, served the purpose of explicating the controller's cognitive work space rather than serving the purpose of representing new ideas for interface design. The domain objects and constraints that usually are mentally considered by the controllers include flight phases, trajectories, fixes, and relationships between them include separations in time. These spatial objects populate in an informal but understandable manner, the sketches.

This sketching process at the same time revealed some of the complexity factors encapsulated in the scenario, which are: (1) the extent of the projection in time, i.e., how much in advance a decision must be taken; (2) the number of conflicts that must be accounted for; (3) the geometry of the conflicts, e.g. two aircraft climbing towards the same point in space from opposite direction. Up to this point it can be said that the sketching process contributed to shape the team awareness of the design problem.

8.1 Switching from user problem to design problem

Sketch in Fig. 8 and 9 report some design ideas following the user problem setting phase. Sketch Fig. 8 represents a possible trajectory visualization strategy for climbing aircraft: The aircraft trajectory is 'connected' to the ground through a vertical surface, so as to make visually salient the aircraft behaviour in relation to the ground in the 3D space, as well as to show the existence of possible conflicts.

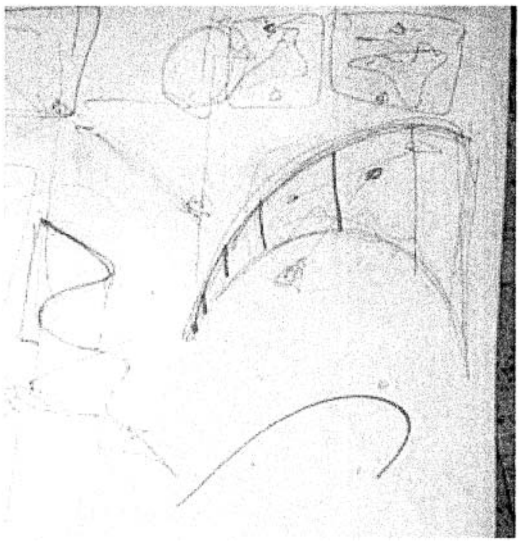


Fig. 49. An early design idea concerning the visualization of aircraft trajectories. In order to make them visually salient it was though to make them connected to the ground

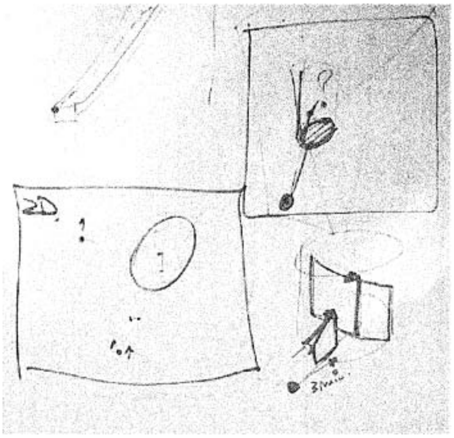


Fig. 50. Combining the 3D representation of trajectory in the 2D radar

Sketch in Fig. 9 continues showing the way a possible conflict would results from this type of trajectory visualization strategy. On the 2D radar (bottoms left in Fig. 9) two aircraft trajectories are expected to cross at some point in time and the controller needs to evaluate the separation in the crossing point. As a first and sketchy solution only a circular area including the crossing point was proposed to be additionally represented in 3D, so to offer the controller an insight into the traffic situation (see Fig. 9 bottom right and the enlarged view of this, in Fig. 10). The cylinder represents a 3D view of the display circled area and contains a climbing aircraft that is likely to intersect a levelled aircraft trajectory. As a result of the previous strategy the aircraft separation is made visually salient by the absence of a joining surface between the two trajectories surface.

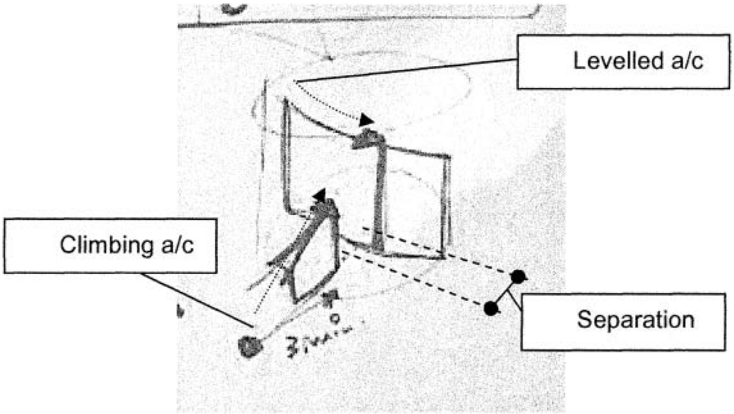


Fig. 10. An enlargement of the previous sketch which shows how the (absence of) separation would be perceived by the user while looking at the traffic scene as reported in the new radar

In this case the absence of a surface was thought to improve the perception of separation. Furthermore, the same sketch also highlights the design problem of switching from a 2D visualization to a 3D visualisation during the interface use. This is a considerable problem due to the necessity for the system user to maintain a perceptual continuity between what is being shown in 2D and the associated 3D representation.

8.2 Focusing on the representation design of relevant user data

It is noteworthy that at this point the amount of design issues was noticeable, and that was reflected by the sketches. These reflected the outcome of team's consideration on data visualization problems (e.g., how to represent the separations? how to make them visually salient) and interaction problems (e.g., how to access the relevant information? how to switch between a 2D and a 3D view?). This last point in particular emerged in this phase and this underlined the exploratory nature of the sketches. However, in order to solve this 'impasse', a design decision was taken and the following discussions focused only on the definition of the emergent display feature relevant to the user task, leaving the definition of interaction details, such as the buttons menus and other functions to be specified later. This approach was indeed supported by existing literature [2,3] and produced sketches like those reported as follows.

Sketch in Fig. 11 provides a display design aimed at supporting controller's decision about evaluation of separations. It provides an enlarged 3D view of two aircraft, represented by the two pyramids, whose trajectories are intersecting. Each colour highlights an emergent design issue that could facilitate controller job, i.e. assess separations. They are:

- Yellow line: projection of aircraft future position, usually made by the controller, e.g., *this aircraft will be there in 2 minutes*;

- Red dotted line: projection of the trajectory of the aircraft, with the actual flight parameters;
- Black line: recent aircraft trajectory;
- Sketched Black line: separation existing between the two aircraft, when Aircraft 1 will be at the intersection point. This would help the controller answer questions such as:
  - *When will aircraft will be at the intersection point in a given time t, which other aircraft will be there or nearby (i.e. likely to be in conflict) at the same time?*
  - *Will be they separated?*

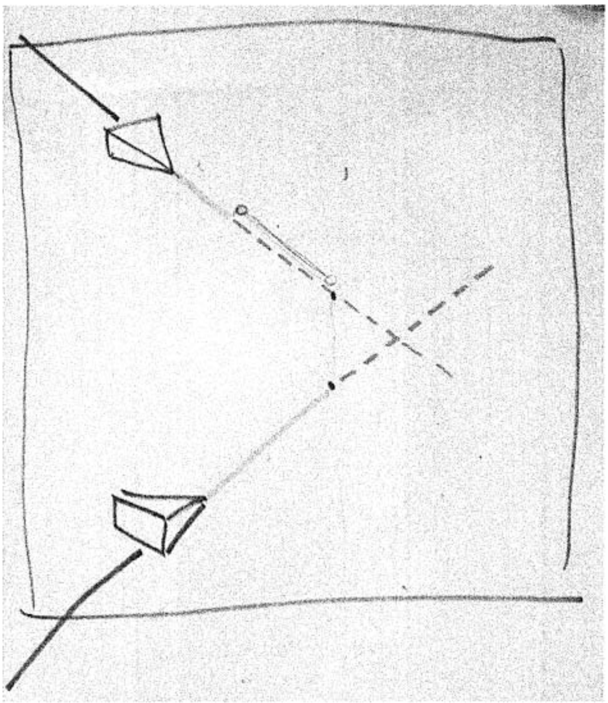


Fig. 11. A representation of the relevant information needed by the controller to evaluate the existence of separations (see text for explanation)

Sketch in Fig. 12 represents the relevant aircraft associated information over time. In particular the upper part contains the following:

- Recent history aircraft information: the aircraft small cubes with a 3D volume are intended to provide an evolution of the short term past trajectories in the 3D space.

- Present aircraft status information: the aircraft, depicted as a little pyramid so to allow an appreciation of aircraft 3D orientation.
- Future task relevant information: This is comprised between the aircraft and the crossing point (indicated in the draft), and includes:
  - Future aircraft trajectory;
  - Position of the aircraft in a given time (Yellow line);
  - (Separation in) time from the intersection point (t);
  - (Separation in) space from the intersection point (s).

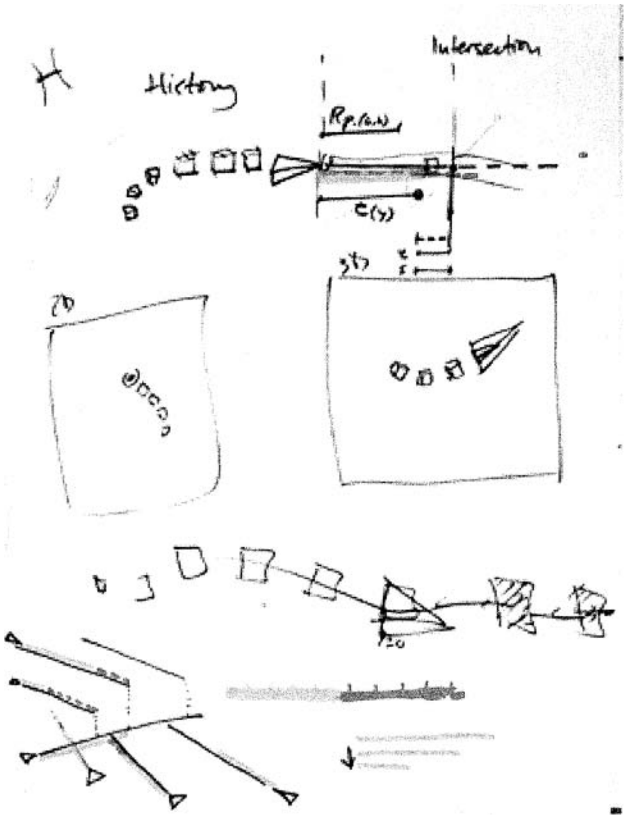


Fig. 12. Intermediate sketches used to define the information to offer to the controller (see text for explanation)

The lower part of the same sketch includes 2D and 3D screens shot of an aircraft in the airfield, so to show how the same aircraft would be displayed in 2D and 3D view. Furthermore it includes on the bottom a view of multiple conflicts where six aircraft are converging towards the same trajectories. Finally it reports also a possible code to distinguish past from future information (yellow/red bar).

A retrospective view on the creative process behind the sketches in Fig. 11 and 12 can reveal some interesting point. Fig. 11 represents an acceptable final draft version of the important task related data to be shown to the user. This is the result of a process where the designers were engaged in discerning the display features among those described in the controller cognitive space, as resulting from the work analysis. Therefore, initially only rough sketches (see Fig. 12) appeared on the sheet representing several possible display emergent features such as aircraft, past and futures trajectories, separations in time, separations in space. These features were progressively visually emphasized, through the use of different colour, or "suppressed"- in other words they went under a sort of tacit selection process driven by the requirements introduced by the work analysis. The sketches reported in Fig. 13 provide another example of this process. At the end of this process important domain raw variables were selected, discussed and combined so to identify integrated emergent task relevant variables.

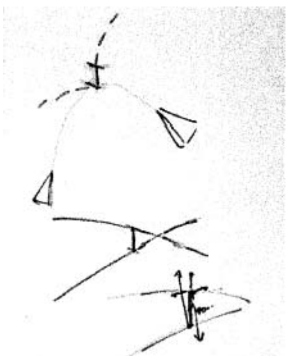


Fig. 13. Two aircraft trajectories intersect. These early sketches represent also raw data as trajectories, vertical separations, crossing point. Which of these to represent? What to highlight? These were some of the question behind this sketching effort.

Looking again at the sketches reported respectively in Fig. 11, 12, and also 13 it could be argued that they have some commonalities with design sketches used in industrial and car design. The latter are based on the basic principle of starting from a first draft sketch, refining what is right, forgetting what is wrong. For example a designer can sketch roughly the shape of a car in perspective, and then can refine its shape a bit, until it is realized that another style is needed for both the front of the car and the rims; thus s/he will use another sheet so as to trace out only the good shape from the first sheet, which then can be completed with a better nose and a new rim style. A similar process occurred in the definition of the user interface objects. As Sketches in Fig.12 and in particular in Fig. 13 demonstrated, several sketches were need to model and refine the display elements, so to allow the emergent variables to emerge on the display surface.

In addition, this comparison with the design sketching technique used in Industrial Design highlights also a difference because the design of the display features must consider the temporal dimension. In fact the objects of the user interface have a behaviour which determines how and when the relevant emergent display features will be displayed. The implication of that is that the sketches are dynamic representation. Furthermore, the temporal dimension assumes a strong relevance especially when it is necessary to define how the user will interact with the interface, as explained in the following section.

8.3 Sketching the Access to Relevant Data

Once the consideration of the main visualization format of the Human Machine Interface was complete, the design team efforts focused on the data access. Sketch in Fig. 14 represents a few screen shots, to be read from left to right. They explain how the user can access the data referred to in Fig.11. On the right, a 2D radar shows a situation similar to the one revealed by the scenario. An aircraft is supposed to flight along a given trajectory. Two groups of aircraft, represented by the black dots, are likely to cross the aircraft trajectory. The controller before authorizing the aircraft in question to follow this path needs to assess the presence of separations in the crossing point.

Thus, in order to evaluate the existence of separation in the crossing point, the controller can click where the aircraft which needs to be de-conflicted is likely to intersect the first group of aircraft in a few minutes (the click area is indicated in the drawing). At this point the data will turn into a 3D representation that shows the evolution of the trajectory in the 3D space and the existence of separation (see sketch in the middle, which represent a climbing trajectory). This representation can then be further enlarged (see sketch in the right, which represent an estimation of the separation, both vertical and horizontal) in order to access the data representation as defined in Fig. 11.

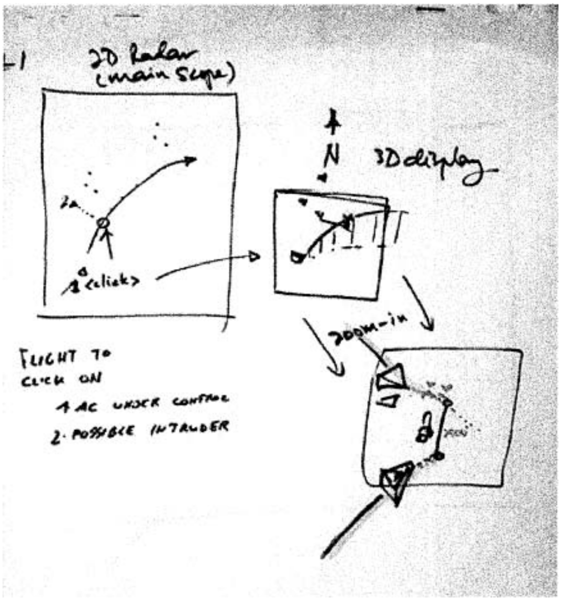


Fig. 14. Interaction sequence in order to access the task relevant data as defined in Fig. 11 (see text for explanation)

Discussion and Conclusions

This paper reported a case study concerning the development of a 4D user interface. It included the user work analysis, an account for the design problem, a framework for considering for supporting spatial-temporal reasoning, and some of the design sketches used in order to move from the finding of the analysis to the early representation forms of the interface. On these bases it is possible to draw a few considerations concerning the role design sketches during this process.

The design sketches presented in this work were focused not only on tentative representations of the display design, e.g. what (and where) information is to lay out, but also included representations of the spatial-temporal variables of the process to be controlled, e.g., aircraft, fix, trajectories, separations, whose combination into a traffic situation determine the controller cognitive work necessary to solve the situation itself. This, in general invites to argue that it seems very difficult to jump directly from the result of a work analysis and the associated visual requirements to the final display without the use of either collective or individual tentative sketches. The aim of these is the exploration and refinement of the proposed display concepts, even before any either formal or informal test with the user take places. For example in this case sketches about the user problem served as a basis for the sketches focused on design issues, and they were developed in an iterative fashion among the designer without any direct contribution of the controllers.

In addition to theses first considerations, it is possible to notice the communicative roles of sketches as they allowed engineers and designer to share a common vision of some user significant spatial-temporal related information, thus confirming the sketching function of externalizing and visualizing problems, as already pointed out in previous work in the architectural domain [12]. The interesting point in this case is that this common vision was constructed around not only design issues, but also process control and user experience issues.

Thus, it might be said that these feed into and delimitate at the same time a sort of design sketching space where the interface was developed. In practice domain variables were drawn in the 3D space. That required some technical skills similar to those commonly used in the product design domain, i.e. perspective construction of a 3D model. Then the design progressed through constant refining of the raw data in order to define the emergent display features to be provided to the user.

In this process the criteria available to refine new concepts depended on the nature of the real user task, i.e. the user need to evaluate a precise type of information (the separations) in a given situation (in case of possible conflict), as well as the nature of the process being controlled, i.e. the specific configuration of air traffic. Finally also the design while being refined introduced new limitations, as the first design decision impact on the following. In conclusions this suggests that the designer/s awareness of these design constraints influences the extent to which the final display design will match with the visual requirements as resulting from the task analysis, thus suggesting in general the need for a design sketching technique for Human Machine interface able to integrate the process, user experience and design constraints.

Acknowledgements

Most of the work reported in this paper has been funded by the European Commission Framework 6 Priority 1.4 Aeronautics and Space Programme "4D Virtual Airspace Management System" Contract No. 012328. We would also like to acknowledge the efforts of our colleagues on the project from NEXT Ingegneria dei Sistemi SpA, ENAV SpA, Digital Video SpA, and Space Applications, Belgium, in this work.

References

1. Endsley, M. R., Smolensky, M. W.: Situation Awareness in Air Traffic Control: The Picture. In M. W. Smolensky & E. S. Stein (Eds.): Human Factors in Air Traffic Control. Academic Press, San Diego (1998) 115–150
2. Cooper, A., Reimann, R.: About face 2.0: The essentials of interaction design. Wiley Publishing, Indianapolis (2003) 86
3. Lauesen, S.: User Interface Design. A Software Engineering Perspective. Addison Wesley, Harlow UK (2005) 167
4. Rettig, M. Prototyping for Tiny Fingers. Communication of the ACM, April, Vol.37, No. 4 (1994) 23
5. Ware, C.: Information Visualization. Perception for Design. Elsevier Science, San Francisco (2004)
6. Wickens, C.D., Gordon, S.E. & Liu, Y. Human Factors Engineering. Longman, US (1998) 226–232
7. Buxton, W.: Sketching and Experience design. Interact 2005, Tenth IFIPT International Conference on Human-Computer Interaction, Rome 11–12 September (2005) 1.
8. Hildebrandt, M., Rantanen, E.M.: Time design. In Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting HFES 2004, New Orleans Human Factors and Ergonomics Society, Santa Monica CA (2004) 703–707
9. Smallman, H. S., & St John, M: Naive realism: Limits of realism as a display principle. In Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting, Santa Monica, CA (2005)
10. Burns, A., Baxter, G.: Time bands in system structure. In Besnard, C., Gacek, C. & C. Jones (eds.): Structure for Dependability: Computer-based Systems from an Interdisciplinary Perspective, Springer. London (2006) 74–90
11. Rasmussen, J. Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering, Vol. 12. Elsevier Science Publishing Co., Inc., New York (1986)
12. Hollnagel, E., Woods, D.D.: Joint Cognitive Systems: Foundations of Cognitive Systems Engineering. FL 33487-2742: CRC Press, Taylor and Francis Group, LLC Boca Raton (2005)
13. Do, E.Y., Gross, M.D., Neiman, B. and Zimring, C.: Intentions in and relations among design drawings. Design studies Vol. 21 No. 5 (2000) 483–503

The Design Sketching Process

Rikke Orngreen
Copenhagen Business School, Denmark, orngreen@cbs.dk

Abstract. Sketching the *what-if* situation provided insight into alternative design possibilities, and gave an idea of how the sketched design will work in the context at hand. Though software provided possibilities of creating a high fidelity executable prototype at a very fast pace, here sketches that were non-executable drawings of the envisioned system, are investigated. The sketches were used in various forms within the development team as well as in collaboration with external experts and users. Where issues of usability and technical requirements are often dealt with separately, this approach suggest simulation through drawings of the context of use and the different users' needs. This turned the design sketching period into both an analysis and interaction design of the new envisioned work processes, as well as gave input to the future production process. The case study illustrating this design sketching process from rough drawings of conceptualisations and detailed storyboards of functionalities was the development of an e-learning platform for case-based learning.

1 Introduction

Sketching provided artefacts that may be used for collaborative exploration of the system being developed, by reading and interpreting the drawings and concurrently (re)write and (re)draw new ideas to the design. As such it created mutual reference points, similarly to mapping models made by users and developers like the rich picture in Checklands soft system methodology [1] and Avison & Wood-Harpers Multiview [2]. In this paper it is illustrated how the process of sketching thus became similar to work analysis, by observing and discussing the sketch and interpreting it, but it moved beyond the present situation, and took the future system's work context into consideration. The challenge was to allow for the design sketch and the design idea it represent to be adjusted easily or even more so, disposed of easily. This process allow for alternatives to be found, as opposed to refining existing ideas, but only if the team makes it an established way of working,

Please use the following format when citing this chapter:

Orngreen, R., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 185–202.

As design sketches were made, worked on, and then either eliminated or chosen, the design of the system was shaped. The transformation from representations of concepts and ideas to representation of functionalities and contents began. This meant a shift from hand drawings of alternative ideas to digital storyboard-like representations of functional requirements, as is discussed in this paper. The creation of such detailed sketches is always a complex and resource demanding task (time, adequate competences and insight into the system are needed). The detailed sketches provide vital visual documentation that can be accessed and refined, and used in the production process. Here, the critical process of trying to anticipate the system being designed, envisioning the future work context, was continued. The difficult part was to recognise when the sketch was a design suggestion (a drawing of an idea) and when it was a visualised specification (a detailed storyboard) similar to a written requirement specification or design documentation.

The case study in this paper was the development of a case-based e-learning platform named CaseMaker. The case-based learning method and pedagogy is often practiced at university level courses. Through the development of CaseMaker it should be possible to develop digital online cases, use them in online and blended teaching situations, and for students to analyse the cases online. As an example, cases in business schools often consist of company descriptions and material (as interviews, balance sheets, project or product presentations etc). Teachers use cases in courses to e.g. show how the different methods or theories work in practice. Students thus discuss the cases using these theories as well as experience from their own work situations, similar companies etc. Like many e-learning platforms, CaseMaker needs to support many roles (that of developers, teachers and students as primary user-roles), and thus also many ways of working with the system.[3]

Design sketching provided the CaseMaker team with design insights about how to work with possibilities from present case-practice. Sketching was also used when trying to get inspiration from other sources and design ideas. This eliminated some of the "childhood" problems that many e-learning platforms were facing. The sketches were used in various forms within the development team as well as in collaboration with external experts and users. Other tools and techniques were also used within the project and in relation to the design, but are not described or illustrated here. E.g. scenarios were used in the very early stages of the project, use-cases and other flow-models in the later design phases, and so forth. The focus of this case study is thus not to describe the big picture of the CaseMaker development process, or to present case-based development, teaching and learning, but rather to

[3] CaseMaker was funded by the e-learning fund at the Copenhagen Business School (CBS). It was and is still being developed at CBS with Rikke Orngreen as project leader and active member of the development team. Primary development team members came from the CBS Learning Lab and included a developer and 1-2 senior consultants who were present at almost every design meeting. During development users and experts were invited to panels, discussing the design, and user workshops were held. Third party consultants were hired, and external programmers worked on the actual production together with an employee from CBS Learning Lab, particularly in the first half year of 2005. During summer 2005 the first version of CaseMaker were finished, but not yet implemented for public use. For more information about the motivation and functionality of CaseMaker, see [3] and [4].

point out specific perspectives on the various forms design sketches may have, and the variety of processes involved.

This paper starts out by illustrating the various forms sketching may take, and the different mindsets, implicit to the way they are used. This presentation (in paragraph 2) is based on very different types of literature, from the semi-abstract to the extremely concrete, and are all discussed with a very practical and specific focus. Afterwards the CaseMaker project is used (in paragraph 3) to illustrate how the design sketching process can use not only one type of sketch and sketching process, but rather an array of sketches that vary through the phases of development.

2 Sketches and their relation to.....

To define a sketch may not seem that hard. The Wikipedia defines (for the time being) a sketch as: "*A sketch is a freehand drawing or other composition that is not intended as a finished work. Sketches usually serve to store ideas for later use....*"[4] There might be slight variations, and certainly the concept of use is broad and varied. It is the overlap with other techniques and the processes involved during sketching that seem to be worth while dwelling on for a moment.

A sketch, a storyboard and a prototype are artefacts that can all be used in relation to the design representation of a software product under development. These three forms of representations have a lot of overlap, but also signify the various stages of the software product – from representations of an idea or concept in a sketch, to functionalities and contents in a storyboard, to detailed design in a possibly executable prototype. They are concepts stemming from different traditions (the creative design and art traditions, the film and multimedia world, and engineering), but can as just described be viewed as part of an incremental and iterative software development process. But just to make the confusion complete; prototyping as a complete software development method (as Rapid Prototyping opposed to the Waterfall Model), both within software engineering, human interaction and software management, almost always contains a sketching and/or a storyboarding period. I.e. sketching then becomes an activity in a prototyping process. Prototyping is often seen as a method that bridges the gab between the various phases of analysis, design and production [5].

Thomas Snitker, a Danish usability practitioner, writes briefly in his newest edition of a book on user-centred design, about advantages and disadvantages of paper prototypes under the heading: "*this napkin is a prototype of my new website.*" [6, page 75]. Now, according to the description of sketches so far, this may lead to the conclusion, that if the napkin contains hand drawn images, it is not a prototype, but a sketch. Both interpretations might be correct, and thus states more about the methodology behind the things drawn than the drawing it self.

[4] The wording: for the time being, is used here to indicate that information found on Wikipedia is not permanent, and is not meant to be, but is rather dynamic in its nature – accessed the November 2005, http://en.wikipedia.org/wiki/Sketch_%28drawing%29

2.1 A design perspective on a sketch

Prototyping, when seen as a life cycle development method that the development team uses from the very first initialization (to begin the development and work on a new idea, a problem or opportunity, or a re-design task) to the test and evaluation of the finished product, might include a sketching period. But to make a sketch is mentally different from making a prototype. The difference was amply illustrated by William Buxton in his keynote at the Interact 2005 conference. Here he presented the difference in paradigmatic views of a sketch and a prototype, which he summarised in a table longer, but similar to table 1.

Table 1: Buxtons table on differences between a sketch and a storyboard, highlights by author[5]

Sketch	Prototype
Invite	Attend
Suggest	Describe
Explore	Retire
Questions	Answer
Propose	Test
Destructive	**Constructive**

That a prototype is constructive in nature, and thus keeps on building on the same ideas and concepts, may not be surprising, but that a sketch is destructive is a somewhat blunt statement (see table 1). The perspective is that many ideas and thoughts may seem good until worked on for a while, and when prototyping the developers may not reach the difficult point of being able to "kill ones darlings". Sketching makes this easier. It is easier to draw something quick on paper, and then realise 30 minutes later the idea was not something meant to be, than using half a day or a week on a detailed storyboard or executable prototype, and then throw it away. The more the idea has been worked on, the more difficult it is to dismiss.

The interesting notion of a sketch as inviting further ideas and suggestions, as concept and thoughts, and not as a product to be further developed, is a fruitful contribution to the design sketch discussion. It first of all fosters the idea that many sketches can and should be made, with some, the majority, being immediately discarded. At the same time, as sketches flow from the developers, the mere process of sketching always seem to open up, rather than close down. Secondly, before a sketch is able provide input to the design process, it also has to be looked at, it has to be read, understood (even misunderstood) and interpreted, before it leads to exploration of alternatives.

At the Interact conference Buxton spoke highly of hand-drawn sketches on paper, but also recognised working processes emerging, which would lead to hand-drawn sketches that are digital. Here he made a point out of presenting sketches that were made by graphical designers and industrial designers that know how to make

[5] Based on notes from the keynote by William Buxton: "*Sketching and Experience Design*", given at Interact 2005, the 14th of September 2005, Rome, Italy. See also Buxtons homepage for more references: http://www.billbuxton.com

beautiful hand-drawn digital images, but also showed how they were "dressed-down" to look like sketches, and to be interpreted as draft suggestions and ideas.

2.2 A storyboard – from movie to multimedia production[6]

Faulkner determines that: "*The initial design for the system can most conveniently be presented in the form of a storyboard*" [8, page 103]. Storyboarding is a method originating in movie production. A storyboard phase typically begins, when a decision to produce a movie is taken. It contains two elements: drawings of the appropriate appearance of the screen, and comments about everything from the set, the actor(s) mood, what should be said, foreground and background action etc. [9,10]

Storyboarding is a method, which is useful in designs that use multimedia, and which have many forms of interaction, because of its ability to visualise, what should take place in the system. Multimedia storyboards typically also contain information about timing requirements, hyperlinks, animations, text etc. [11,12]. A storyboard can serve as a tool, which supports the design process in conjunction with scenarios [13,14]. It can also serve as a way of documenting the decisions taken, and then later be used when the production of scripts, media production and authoring (programming) begin [7,15,16]. Figure 1 shows an example from Hofstetter's book on multimedia literacy, and even though this book only gives an extremely short introduction to storyboards, Hofstetter is one of the few who show an example of a format that can be used for documentation [15].

In Orngreen and Pries-Heje 1999 the storyboards from various projects are also depicted and discussed. A resulting framework for contemplating the degree to which storyboards are continued to be used throughout the development process is given. The idea is not to promote the use of a one and only storyboard standard, but rather to contemplate the need for storyboards based on 4 variables of the system under development: complexity, size, amount of re-use of material, and media usage. The need to make detailed storyboards for every part of the system grows with: high complexity, large systems, large amount of material (contents, navigation etc.) being made from scratch, and high degree of media usage. Whereas simple small systems, with a lot of reuse and little media usage just need conceptualisation of the general idea of the system [12]. This is to ensure both an economically sound approach, and a user-centred approach that matches the envisioned work context.

[6] Large parts of this paragraph draws on previous analysis and work on storyboards found in Rikke Orngreen's phd. dissertation on the development of multimedia teaching cases [10]

Module: _____ Strand: ________
Filename: ________________
Screen No. _____ of ________
Images: ________________

Audio: ________________

Video: ________________

N A V I G A T I O N
Next: ________________
Back: ________________
Menu: ________________
Help: ________________
Notes: ________________

Figure 1: Storyboards, design and documentation tool. Source: [15, Figure 45-6, p. 379]

2.3 Sketching in user- centred and participatory design

Sketching is also used to understand present practice with a strict analytical focus, rather than the more analytical creative / design focus of the two previous paragraphs. Visual mapping of the present context, rich pictures, geographical models etc. are examples of these [1,2,23]. They are either drawn by the developers during or after field studies or the users themselves in a more participatory oriented development process. E.g. the more detailed a storyboard becomes, the closer it resembles the prototype as a product, and the farther away it moves from the Buxton idea of a sketch. However, both a storyboard and a sketch in Buxton's notion have common perspective in that the sketching of ideas and functions are primarily done as part of the developers work. In user-centred design the approach is to involve users and clients in the sketching work of present practice and the vision. [13,17,18,19]

Adaptations of storyboards are used to interpret the current or past situations. Movie analysts use them when interpreting scenes from movies, describing and showing areas that are not immediately in the frame of the camera, but vital to the story. (As something physical or a sense of atmosphere). In IT development, drawing flows of frames can be used with annotation strategies from dance notation and video analysis literature to note important contextual features, feelings, and dynamic behaviour (user actions) that the system under development should adhere to. [7].

The vision phase is one of 4 phases in the participatory framework for pre-investigations, named the MUST method by Boedker, Kensing and Simonsen. The techniques used in this phase of exploring the vision of the system to be developed are: virtual mapping, drawings, collage of sketches/ photo montage and scenarios. In the presentation of the MUST-method, the questions raised by the user-groups when appying these techniques are of special interest. Questions as: what if there is a new system which is able to...; what if, we organise ourselves differently etc. [17]. These are questions and answers that will be reflected in the sketches. According to Boedker, Kensing and Simonsen, a pre-investigation workshops that make hand-drawings, collage, design sketches and data models can use these tools and techniques for dialogue on abstract knowledge of present and new IT work practice. [17, figure 8.2 page 216 and 270-319]. Their starting points are very oriented towards "problems" in the present work practice and "solving problems" with IT support, and not so much towards opportunities and exploring, and experimenting with innovations. It is however, interesting to see, how this starting point joined with these tools, provide a sum of artefacts of various kinds of sketches that creates a solid interpretation of the present as well as envision of the new. This is a solid basis for the development team to work with after the pre-investigation.

Similarly development teams with roots in user-centred design and human computer interaction often invite users and clients to see various representations of the teams understanding of the context and the teams design suggestions. E.g. Buur and Boedker describes the design collaboratorium, where users are invited to workshops contributing with reflections on the teams design suggestions (often prototypes, but also collages of other users' statements, sketches etc..) [19].

2.4 Sketching software tools

Newman, Lin, Hong & Landay identifies the design process as consisting of phases of: Discovery (where hand-written notes are primarily used); Exploration (where hand-written site maps and storyboards are used), Refinement (where digitised schematic and mock-ups are used); and Production (where digital prototypes are developed) [14, see for example figure 4, page 274]. In their wording site maps and storyboards are both visual representation of the system structure and flow / navigation, as opposed to also having storyboards that contain information about contents and functionalities, as discussed earlier.

With the objective to develop a tool, named DENIM (Design Environment for Navigation and Information) that assist web-designers, Newman et al carries out a thorough qualitative investigation of how 11 web-designers work [14]. They find that all of them use "*low-fidelity sketches*" on paper, which are important for the design process focusing on structure and navigation and are means of communication (between developers). However, they also find that designers are reluctant to show them their drawings, and that they would never show clients these rough sketches, only to co-designers, who provide valuable feedback early in the design process. If a sketch idea is presented to clients, digitalised "nice" version of the same idea is made (in Photoshop or Illustrator) [14]. Similar conclusions and objective of the two tools (FreeForm and SUMLOW) are discussed in Plimmer &

Grundy 2005. They look at how hand-drawn sketches made with stylus on tablet PC's can be automatically "*beautified*" into resembling real data models, database entry forms etc [20].

3. The CaseMaker design sketching process

Numerous examples of sketches from the CaseMaker project could be presented here. From visual maps of the existing context, over heaps of paper sketches of design suggestions, to many, many PowerPoint files of storyboards. Three examples have been picked out to show the stages of the design sketching process and to highlight some of the problems and opportunities the tools encountered, both on a specific level for this project and as general lessons.

3.1 A case of a technical function that initiated an overall design strategy

In the very first "whiteboard discussions" in the core development team (November 2003, just after receiving funding for the project) the team found that the experience with not only CBS' e-learning platform (which is Sitescape), but also experiences gained as invited teachers at other universities or corporate business universities, all pointed to the need for more flexible user / role assignments. In many platforms the administrator has to (semi)manually allow for users to enter the system, by drawing on information from the student and employee registration databases (known as LDAB). The registration database is updated, but not nearly fast enough, and even the guest-ldab gives problems. In this way the first whiteboard sketches served as a way of reflecting on present practice. These were very technically oriented problems, that are usually not dealt with at this stage, but which have severe influence on the users' interaction with current e-learning systems, and consequently the users' experience with and feelings for the e-learning system. It led to envisioning new ideas that were very different from present practice, and pointed towards more open strategies of allowing users to use the system if they wanted to, whether they were officially CBS students or not. This brought the context back into play, with questions like: "if we make it very flexible and user-controlled, what about security, and will the CBS Management approve of such an approach?"

The whiteboard flow-sketches of signing in took many forms. Figure 2 shows a simple tree-structure, which at this time of the discussion was enough to trigger a very long dialogue and many more sketches and other tree-structures. Together with issues about encouraging developers to share material and cases, the above issue eventually led to the decision to go for a very open standard system. It could later even be rolled out as an open source project. One might say that the discussion on the log-in procedure was the initiator to the design paradigm and mindset which was applied throughout the development period.

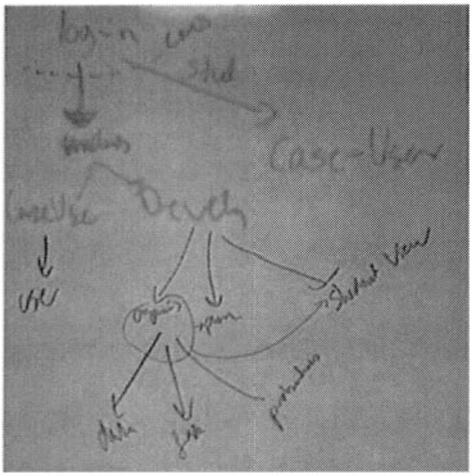


Figure 2: Whiteboard flow-sketch, log-on procedure led to open standard decision (from November 2003)

3.2 Trying to escape the prototyping cravings and software tools that came too early in the process

A year later the more detailed process of getting the structure and contents of the whole system in place began. At this point the contents and functionalities were clear, but the design of the structure, how to interact with the system, and which paths to follow had not been discussed yet. It is noteworthy that there were large breaks in the development process, and that the time difference of 1 year reflects these breaks. Probably approximately 2-3 months would have gone by, had the development team worked full-time. The objective now was to consider, how the structure should be, so that it supported a natural work process for the users. Though sketching was done in more details with other aspect of the system, taking point of departure in the log-on discussion from before gives basis for comparison.

A member of the team and an e-learning and developer consultant wanted to apply a software tool for sketching. At first the DENIM sketching tool mentioned earlier was used, but it sort of put the developers in a too rigorous and detailing state of mind, compared to the kind of switch between reflection and designing needed (see figure 3). When using DENIM (which is a newer edition of the first sketching tool by primarily Landay and Myers named SILK [21]) sketches often became low-fidelity prototypes with focus on structure. I.e. even though it visually were rough designs (figure 3), as developers we tended to treat these as design suggestion to be critiqued and commented on in a way that nourish further development, and not so much to rephrase the idea all together. As a consequence, DENIM was abandoned.

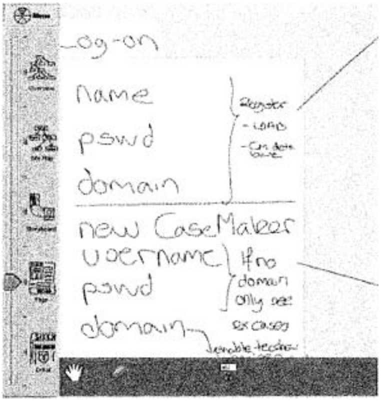


Figure 3: DENIM flow-sketch that was newer finished (from November 2004)

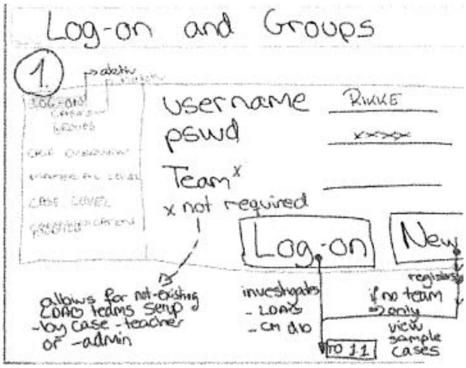


Figure 4: Journal note sketch that was transformed to a similar paper process (from November 2004 – cropped slightly)

Then, there was a brief attempt to still use the tablet PC as "paper" and the Microsoft windows journal software with a complete blank page as starting point (figure 4). However, again the idea of being fast at sketching an idea, read and interpret it, then quickly thereafter present the next idea, and then make the next etc.... seem to come second to the notion of getting more and more details into the picture, ones the software comes into play. This is fine if the development team is aware of this "hook" to detail that the software gives. As with the DENIM attempt the Journal sessions only lasted two days with a couple of hours work each day, and were then quickly discarded as not usable at this stage in the process – at least for this kind of system. The development team members had had similar experiences on previous occasions. [7] The experience is that the software is very appropriate for late detailed design and refinement stages of a design process, but not for contemplating the surrounding context of users and for getting into the "sketching-mode".

3.3 Getting a user-centred and collaborative approach, by meeting and simulating context to eliminate and refine choices

The design sketching process is also about willingness to view design from the system or the user perspective. Where system developers in the software engineering tradition do have the users in mind, it is often the needs of the common or average use(r). In the rational unified process for example, use-cases are applied to show the users a way through the system. This creates a basis for defining system requirements [22]. These user sketches and descriptions often become self-fulfilling prophecies, as the user acts in ways that are intended, illustrating how useful the system will be.

[7] The experience with DENIM in particular and using PC (tablets and ordinary) in general, stems from both the CaseMaker project during 2004 and 2005, but also from several student projects, where DENIM was used on real companies' design and re-design tasks given to students.

In a user-centred approach, it is the various and individual contexts of use that are in focus, and the often contradicting ways users act [23]. Sketches and descriptions thus have to take point of departure in: Why and how can users use the system? Rather than only contemplating how the system can support the user? Rephrasing this to the CaseMaker design process: the software engineering approach (to put it a bit squarely) deals with: how can CaseMaker provide cases to case developers, teachers and students? And the user-centred approach deals with: how would different users work with CaseMaker (not only a developer, a teacher and a student, but various developers, teachers and students).

In the beginning the CaseMaker project team concentrated on making storyboards fulfilling needs that had been observed in case situations. E.g. the two sketches in figure 5 are analytical sketches or storyboards (see 2.3) of students working with traditional and multimedia cases. (From Orngreen's previous work on teaching cases [7]). The sketches are part of an analysis of a video recording. In addition, the sketches are supplemented with a larger written interpretation). I.e. first the sketches are made, and afterwards they are read, providing reflections that are documented in the written interpretations (the written interpretation is not shown here.[8] The in- and out-of-frame concepts are seen on the sketch, where the square denotes the frame of the video screen and the drawings out-side of the frame documents what is not visible to the eye.

Such knowledge of existing case situations supplied input to discussions on e.g. types of cases that users would prefer to work with. Some case developers have material for cases that are appropriately presented to the students in a very linear fashion, others in a more networked fashioned, others again in a hierarchical tree structure. The teachers may want to give the student some material at the beginning of a case teaching event and additional material later on. Students would most probably like to access material at different times, but still be able to work collaboratively, leave notes to each other etc.

[8] In some analysis the hand-drawn analysis was made on top of screen-dumps of the video recordings. However, in this situation the students wanted to by anonymous. Also, similar sketches are sometimes made during the observation not using video recordings, as video-observation and analysis is a very time consuming, and thus expensive process to carry out.

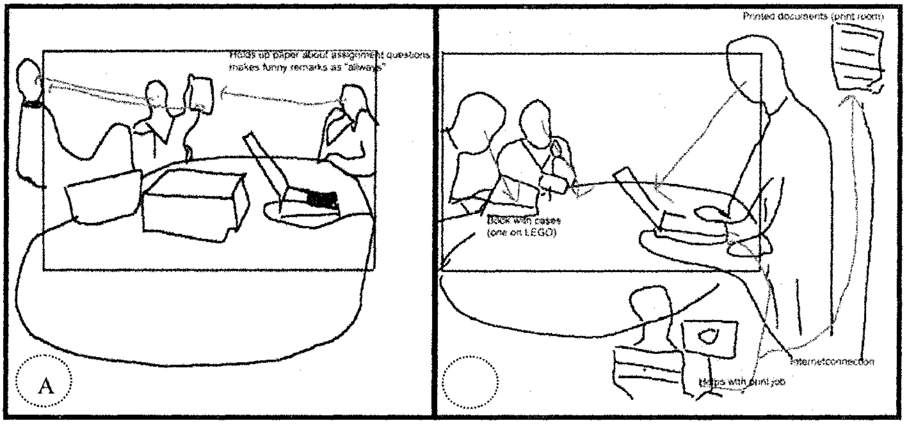
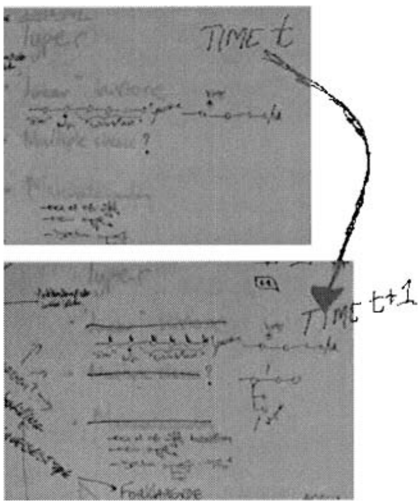


Figure 5: An analytical sketch of students working with cases, from [7, the studies were conducted during 1999-2002].

The start-up meeting in July 2003, mentioned earlier, was a two day workshop just for the development team (4 persons on this occasion). Here we also used the whiteboard-sketching sessions, to draw on each other's sketches. As seen on figure 6 there were several ideas and discussions about the types of cases at time t. As the discussion went on, some were eliminated at time t+1. The different words used to illustrate different ways through the case material were not easy to understand between the members, and more diagrammatic notations were then used. The diagrams led to an understanding that several types could be encompassed in the same notion of a case (se figure 6).

Figure 6: Whiteboard-sketch – picture from two subsequent periods (July 2003).



At some point we contacted experts and potential users (teachers and students). Panels were held (with presentation of ideas and storyboards and following discussion). Later vision workshops [17] and interviews about CaseMaker were also conducted. In these panels and workshops present practice, visions of CaseMaker and specific functionalities were either reinforced or dismissed. E.g. the idea of letting CaseMaker have cases that were sort of simulated story lines with multiple choice branching (i.e. the story would go in different directions depending on the choices) were eliminated. This was done because though simulation-based learning is interesting, it was not seen as part of case-based learning, but quite different from it. It would require a complete different mind-set compared to the other features included in CaseMaker, and might therefore seem more distracting and off-key rather than part of a coherent platform.

As a result storyboards as seen in figure 7 were made to provide estimates of the necessary time and resources needed for the programming phase, as a means for considering choice of coding platform and structure, as well as making the more detailed user evaluations (the user workshops as opposed to the user and experts panels). The example is still the case types, but now instead of referring to case types, the storyboard refer to case structures, and allow the user – the case developer – to structure the case material into headings at the same time.

In these storyboards (figure 7) CaseMaker was always shown as having a pane-menu, though the format had newer been discussed. Just prior to the beginning of the programming process, it turned out that the members in the development group had different thoughts on why the pane-menu was used. Some thought the format was chosen to show overview of where we were in the system when dealing with a certain function. That it was not at all part of the design chosen and this was still to be determined. Others saw the storyboard as representing the final interaction design with respect to navigation. (Not with regard to colours, font etc., which were deliberately made ugly and dummy-like to show the state of the system, i.e. that there was room for re-design).

The team decided to investigate appropriate navigation options, by sketching and simulating use on a more detailed level. The team would imagine different paths through the system depending on different use objectives and preferences (see figure 8). Also, small pieces of papers with various input and contents to cases were made as to simulate different work processes, similar to the methods described in the many articles on evaluating paper mock-ups. (Including methods dating back to the often quoted "design at work" [24] and newer text books on interaction design [13] etc.)

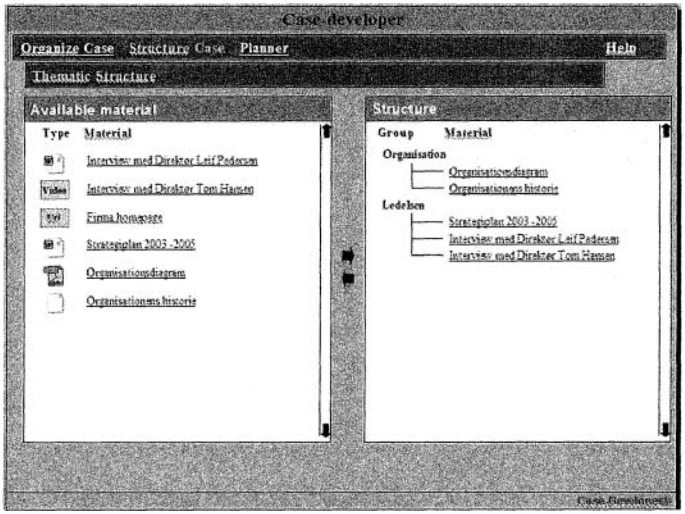


Figure 7: PowerPoint storyboard, case types in the case-develop module, from February 2004

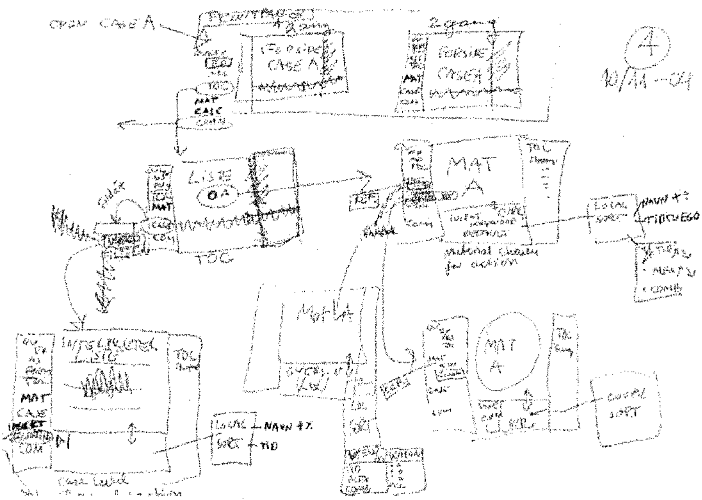


Figure 8: Paper sketches, trying out various work processes, November 2004

Up till then our system in the storyboards resembled a stand-alone system, and not an open flexible web-platform. Though the functional possibilities may be the same, the mere "look and feel" of a system, of the user's perception of it, also have great influence. Furthermore, the pane-menu made an overview of the features in the system difficult, and created a laborious work process. The shift in interaction form only resulted in some extra hours of dialogue and evaluation with users. It would have been difficult to change, had the system been implemented that way.

Figure 9 shows the appearance of the first version of CaseMaker. The example chosen are the case-developer module and the case type's possibilities, now referred to as different case structures. The figure shows that the pane-menu was abandoned (figure 9).

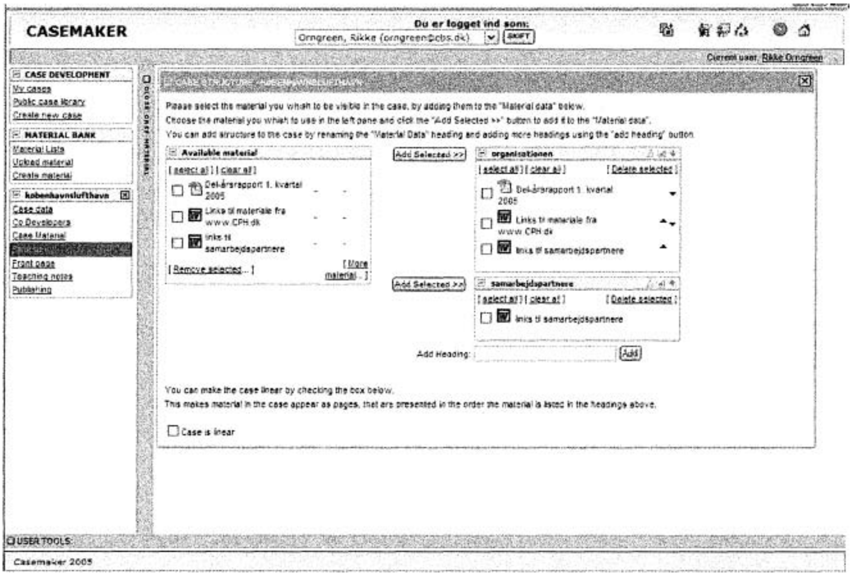


Figure 9: the first version of CaseMaker from summer 2005

5 Learning from the design sketching processes

Sketching is often seen as a very individual creative process of an artist and designer making a piece of art, a movie or a new product. The sketch becomes a medium for an internal dialog of suggesting design, and getting to new design suggestions by investigating their application (reading and interpreting the sketch), and by using the sketch to reflect on advantages and disadvantages. This process is also fully convertible to a collaborative process. However, the individual creative process must not be underestimated, as the collaborative design sketching process can easily lead to overruling of ideas that might have been great (but could be too quickly eliminated by other members of the team). As in the CaseMaker project marked by long periods of only periodical work on the project, sketching is a quick way of getting into the system "idea" again or introducing new people to the concept, by showing old sketches, and drawing new with them as you speak.

Getting people to sketch is not always easy, and it can be difficult to escape the "prototyping mode" of wanting to construct rather than being open and inviting at the same time as being destructive (table 1). In the project team many were used to a process of getting ideas, programming them, and then refine, rather than simultaneously looking at possible alternatives. It is a mental shift to adhere to the design sketching process and this is not an easy shift to make.

Both the CaseMaker case study and literature show it can be difficult to "kill ones darlings". The perspective here is that recognizing what the system is *not* about, is also a step forward, particular if this finding can be made early in the design process.

The CaseMaker design team also wanted inspiration from other solutions to come forward very early in the process. E.g. research on existing e-learning platforms demonstrates the difficulties in getting an overview of the many, many postings students make during a session (a course, an assignment or now a case). For teachers and students alike it becomes difficult to get an overview of the learning process, and it is not a matter of applying simple search-functions, but rather to get an indication of: who always begins a discussion, who makes summaries, which topics from the curricular are prioritised etc. Sketching on basis of other types of applications addressing this difficulty became a way forward. Particular qualitative data analysis applications were inspirational as they use visual representations. Sketching these features in a CaseMaker setting has lead to design suggestions with contents related visual overviews of weighted, dynamic and annotated relations in various spatial and time dimensions. These features also inspired general consideration on what is now called proactive teacher tools [4]

The design sketching process as discussed and carried out in the CaseMaker project is much about user-centred design and human computer interaction, but at the same time the CaseMaker development team had the objective to allow for the more technically oriented perspectives to be discussed. By only using written documentation and very formal modelling and drawing techniques (data models, flow charts etc.) the interaction with the system and the intended feeling of working with the system, which are a vital part of the work context, become blurred. Sketching created a basis for a dialogue, and also provided means for simulating the work context in the design process.

Acknowledgement

The CaseMaker project is funded by CBS, through the E-learning Fund. The first version of CaseMaker was developed at CBS as a joint venture between the author at Department of Informatics and the CBS Learning Lab, in particular Kenneth Andersen and Claus Nygaard. Several external partners, consultants and programmers have worked on the system, and Mads Syrak, from Syrak data as well as Karin Levinsen, a practitioner and researcher from the department of Informatics have both played vital roles as well. Thanks to all and everyone!

References

1. Checkland & Scholes (1990): *Soft Systems Methodology in Action*, John Wiley and Sons.
2. Avison & Wood-Harpers (1990): "*Multiview*", Blackwell

3. Orngreen, Rikke (2004): "CaseMaker – An Environment for Case-based e-Learning" in *Electronic Journal of eLearning, EJEL*, vol 2, nr. 1, p. 167-180
4. Orngreen & Levinsen (2005): "Proactive teacher tools for online teachers", in *The 8th IFIP World Conference on Computers in Education (WCCE2005)*, 4-7 july 2005, University of Stellenbosch, Capetown South Africa, 2005, 10 pages
5. Sommerville, I. (1992): *Software Engineering*, Fourth edition, Addison Wesley Publishing Company, USA
6. Snitker, T. (2004): *Breaking Through to the Other Side,* Nyt Teknisk Forlag, Denmark
7. Orngreens, Rikke phd. dissertation on the development of multimedia teaching cases
8. Faulkner, C. (1998): *The essence of Human-Computer Interaction*, Prentice Hall.
9. Bordwell D. and Thompsen K. (1997): *Film Art, an Introduction,* McGraw-Hill, 5. edition
10. Hart J. (1999): *The Art of the Storyboard - Storyboard for Film*, TV and Animation, Focal Press
11. Landay, J. (1996): Interactive Sketching for the Early Stages of User Interface Design, Ph.D. Thesis, School of Computer Science, Computer Science Division, Carnegie Mellon University, Pittsburgh, PA December 19, 1996, CMU-CS-96-201 Also appears as CMU-HCII-96-105.
12. Orngreen, R. & Pries-Heje, J (1999): "Designing the design - need-based storyboard adaptation for multimedia systems design" from Proceedings of the 22nd Information Systems Research seminar in Scandinavia *(IRIS22):* Enterprise architectures for virtual organisations, 7-10 August, 1999, Keuruu, Finland. Vol. 3
13. Preece, J.; Rogers, Y. and Sharp, H. (2002): *Interaction Design – beyond human computer interaction*, John Wiley & Sons, Inc., New York, New York, USA
14. Newman, Lin, Hong & Landay (2003): "DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice", in *Human-Computer Interaction,* Lawrence Erlbaum Associates, Inc, .2003, Volume 18, pp. 259–324
15. Backer, D. (1994): "Multimedia Presentation and Authoring", in Buford, J.K: *Multimedia Systems,* Addison Wesley, Publishing Company, ACM Press, SIGGRAPH series, New York, New York, USA, pp. 285-303
16. Hofstetter, Fred (1997): *Multimedia Literacy*, second edition, The McGraw-Hill Companies, Inc.
17. Simonsen, Kensing & Bødker (2000), *Professionel IT-forundersøgelse* (ENG: professional IT pre-investigations), 1. udgave, Samfundslitteratur.
18. Holtzblatt, K. and H. Beyer (1996). "Contextual Design: Principles and Practice." in *Field Methods Casebook for Software Design.* D. Wixon and R. J. New York, Wiley & Sons: p. 301-333.
19. Buur & Bødker (2000): "From usability lab to "design collaboratorium" - designing with interaction styles", in *proceedings of Designing Interactive Systems*, DIS2000, New York 2000
20. Plimmer & Grundy (2005): "Beautifying Sketching-based Design Tool Content: Issues and Experiences" in *6th Australasian UserInterface Conference* (AUIC2005), Newcastle, Vol. 40. M. Billinghurst and A. Cockburn (Eds.), 8 pages.
21. Landay & Myers (2001): Sketching interfaces: Toward more human interface design." In *IEEE Computer*, 34(3), 56–64.
22. Conallen, J. (2000): *Building* Web Applications with UML, Addison-Wesley Longman, Inc., reading, Massachusetts, USA

23. Nielsen, L (2004) Engaging Personas and Narrative Scenarios - A study on how user-centered approach influenced the perception of the design process in the e-business group at AstraZeneca-, CBS Ph.D.serie, nr.2004-17
24. Greenbaum, J. and Kyng, M. (editors) (1991): *Design at work - cooperative design of computer systems*, Lawrence Erlbaum associates publishers, New Jersey, USA.

Principles and Practice of Work Style Modeling: Sketching Design Tools

Pedro Campos and Nuno Jardim Nunes
LabUSE – Laboratory for Usage-Centered Software Engineering
Dep. of Mathematics and Engineering, University of Madeira
Campus Universitário da Penteada, 9000-390 Funchal – Portugal

Abstract. In our research, we have been combining Work Style modeling with the well-established principles of Usage-Centered Design having the objective of designing and evaluating better design tools. Our approach distinguishes itself from the fact that it combines work style quantitative data (easily obtained through logging tools) with qualitative data that predicts a given tool's level of acceptance. We describe a set of principles that were proven successful during this design process, illustrate sketches of the tools, and highlight the relevant design aspects that worked and those that didn't work.

9 1 Introduction

Trying to promote a better understanding of the relationship between work-domain based experimental studies and iterative design of prototypes can only be achieved by designing new methods and models that are more useful and usable. We have designed a framework and a model (that we call the *work style model*) as an aid to the complex task of designing tools that support the equally complicated task of interaction design itself.

The work of interaction designers is very multifaceted: designers need practical guidance (not canned solutions), as well as concrete principles. Good models and good tools should highlight opportunities for innovation, leave the details open (concentrating on essentials), invite creative projection and inform – and guide – towards good design. Although some research has been dedicated to examining why modeling tools are not used [1] or creating frameworks that can measure the tools usability in a cost-effective way [2], our approach distinguishes itself from the fact it combines work style quantitative (easily obtained through logging tools) with qualitative data that predicts a given tool's level of acceptance.

The remaining of this paper is organized in the following way: Section 2 briefly describes related research on work domain analysis, and models and tools for the interaction design tasks. Section 3 defines a new framework for studying the work of interaction designers as well as our work style model. It also describes the foundations of the framework as well as why it was designed that way. Section 4 presents the models and sketches of the design tools we have been building and evaluating (namely CanonSketch and TaskSketch). During this process of design and evaluation, we have collected and summarized a set of heuristic principles that

Please use the following format when citing this chapter:

Campos, P., Nunes, N. J., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 203–219.

proven useful. Those principles are presented in Section 5, which also describes which design aspects worked and which design issues didn't work. Finally, Section 6 briefly outlines our conclusions.

10 2 Related Research

Human Work Interaction Design is a recent research area which was born under the auspices of IFIP's Working Group 13.6. In a recent workshop [3], new themes and directions of research on human work analysis and design to support it have been outlined. The main target of the work group is the *analysis of* and the *design for* the variety of complex work and life contexts found in different businesses. Software design (which includes interaction design) is often a team activity and most projects involve stakeholders with different backgrounds that must cooperate in many different and interrelated activities, so it fits very well into the Working Group's directions.

Complex work activities increase the difficulty of predicting the level of acceptance of novel technology and how it will be used in practice. An important and open research question is how to translate usability evaluation results into concrete design decisions [4, 5].

The Technology Acceptance Model (TAM) developed by Davis and colleagues [6] is a widely used theoretical model in the Management Information Systems (MIS) field. Basically, it attempts to predict and explain computer-usage behavior, offering both researchers and practitioners a direct, pragmatic instrument to measure a technology's degree of acceptance. Morris and Dillon [4] pointed out that TAM offers HCI professionals a "theoretically grounded approach to the study of software acceptability that can be directly coupled to usability evaluations".

Wu and Graham [7] describe a novel model for recording the working style of people using an interactive system. Workstyle modeling complements task modeling by providing information on how people communicate and coordinate their activities, and by showing what style of artifact is produced.

The workstyle model was developed in the context of the Software Design Board project, a project aiming to provide better tools for software design. The model was validated through evaluation of existing design tools, and motivated the design of a new software design tool. It is comprised of eight axes: four of them describe collaboration style (Location, Synchronicity, Group Size and Coordination); the remaining four describe the nature of the artifact being produced (Syntactic Correctness, Semantic Correctness, Archivability and Modifiability).

The workstyle model for software design has the advantage of being simple to apply and clearly showing where a tool can fail to match the intended work context. However, it is not sufficient for capturing UI specific activities. Transitions (or shifts) in the workstyles of interaction designers are more frequent and more intense than in any other software design activity. This was the starting point of our investigation towards building a framework for interaction design activities, which we will describe in the following Section.

11 3 Styles for Work Styles: Definition of the Framework

Interactive Systems Design methodologies, such as [8], often describe users in context by using the concept of *actors*. Usage-Centered Design (UCD) [9], for example, separates the actors of a system from the *roles* they play during the system's usage. Indeed, users adopt several roles during the usage of a system, just like film actors do, but they also switch roles throughout that usage. Although interaction design methods are well conceived to realize systems supporting the roles of usage, few methods provide support for flowing from different contexts/needs of usage.

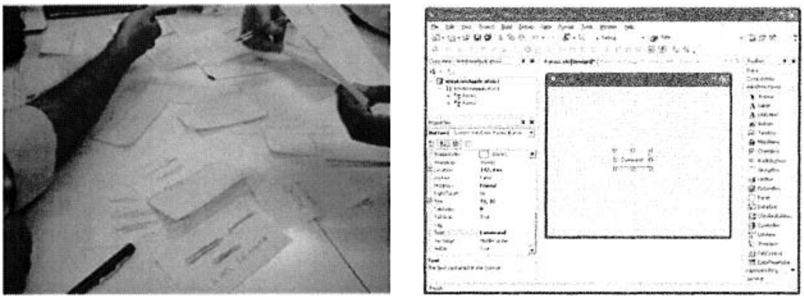


Fig. 51. A Work Style Transition: working in groups using low-tech materials for task modeling and clustering (left). After task modeling, each team member is assigned a set of tasks and builds concrete prototypes supporting those tasks using a visual interface builder (right).

Interaction Designers are the users of Design tools themselves, and in this context, we have developed and applied a model for describing the contexts in which they work, by modeling their *Work Styles*. A *Work Style* is an informally-defined set of values in *n*-dimensions. These dimensions describe the most important aspects of the way users work in order to achieve their tasks. A work style *transition* (or change) is a change in one or more values of a work style. A region (or plane) in a work style model is a set of work styles. Systems supporting work style *regions* are systems that can adapt to and support transitions in the users' styles of work. Figure 1 shows an example of a work style transition in the life of an interaction designer: on the left, a team of developers works together using post-it notes for task clustering in a spatially useful style. After this, the team splits and each designer is assigned a set of tasks and builds a concrete mock-up of the interface using an interface builder. Each designer *transitioned* from a low-detail, collaborative, low-tech work style to a high-detail, high-tech, individual work style.

3.1 The Work Style Model

The eight continuous axes in our Work Style model for UCD are shown in Figure 2. These axes are grouped under three main categories:

- **Notation** style-related dimensions (Perspective, Formality and Detail),
- **Tool usage** style-related dimensions (Traceability, Functionality and Stability) and
- **Collaboration** style-related dimensions (Asynchrony and Distribution).

Each of these dimensions is described in [10], and for each dimension there is a set of questions that can act as guidelines aiding the process of work style classification. In this section, we briefly describe each of these dimensions and provide a set of questions that can act as guidelines to apply the model to tools, notations or, in general, styles of work adopted by interaction designers.

Perspective. This axis plots the perspective, or view, of the artifact being developed. **Questions**: is the notation capable of expressing business goals? Or non-functional requirements such as customer experience requirements? Does it help define the purpose of the system? Does it describe interaction aspects of the system? How close is it to the final product?

Formality. This axis classifies the work style of a designer creating artifacts in a formal vs. informal way. In the early stage of the process, designers use rough, ambiguous sketches to freely express ideas quickly [11]. This work style also fosters comparison of design alternatives and creativity, since the uncertainty of sketches encourages the exploration of design ideas. As design progresses, a more formal style of work is incrementally adopted, as designers need to focus on the precise meaning of their models. An example of this shift is moving from a whiteboard to a CASE tool.

Questions: how easy is it to define rough ideas? Does the meaning matter? Does the notation force you to use a rigid syntax/semantics?

Detail. We added this axis to plot the level of detail (or abstraction) the designer is working at. High-level, abstract models facilitate problem solving in organization, navigation and overall structure of the UI, leaving aside the details. On the other hand, realistic (or figurative) prototypes address high-detail design issues [12]. Disciplined designers tend to assume a work style that goes from higher-level abstract representations towards more realistic and detailed representations as the process evolves [12].

Questions: can you abstract irrelevant details using the notation? Can you think about navigation and structure of the overall interaction using the notation? Can you incrementally add enough detail?

Stability. This dimension describes how difficult/frequent it is to modify any aspect of the artifact(s) being developed. A content inventory of the UI modeled in a UML tool is highly modifiable because it is easy to change names, positioning, size and other aspects of the elements. This is opposed to drawing a model of the UI with pen and paper, since changes are harder to accomplish. Brainstorming, for instance, is a very unstable work style because changes are very frequent. High values in this axis indicate less frequent or less significant changes.

Questions: How easy is it to modify previously created artifacts using the tool? How frequently do you make those changes? Are there particular changes difficult to accomplish with the tool?

Traceability. This is a new dimension we introduce. It describes if the elements of the artifact being developed are consistent and interconnected (thus being highly traceable) or if they are completely unrelated and independent. As an example, developers might adopt a work style in which they choose to keep links from task cases steps and the concrete UI widgets that implement those task steps. In this case, it is possible to trace a task step to the concrete widget and to trace a widget to the task step it implements. This dimension is closely related to stability and the number of artifacts produced during a project. As they increase, traceability becomes more important.

Questions: Are you using the tool to maintain interconnections between model elements? How important is it to navigate through your model? Does the tool maintain several different views in a synchronized way (e.g. design view and code view)?

Functionality. This is also a new dimension we introduced. It represents how much functionality is being addressed (by using the tool to build a prototype). There is a barrier between software engineers and usability professionals regarding this matter: software engineers are engaged into building reliable, functional systems, leaving user-friendliness to the usability specialists. Usability and interaction designers, on the other hand, first design and test the interface with end-users, leaving implementation to software engineers, regarded as functionality builders. Those two processes should not be separated [13] and considering this dimension will help overcome that barrier. This dimension is also important because designers combine visual design (presentation issues) with interaction design (behavior issues).

Questions: How much functionality, behavior and dynamics can you add to your prototypes using the tool? How easy is it to test the interaction by using the tool?

Asynchrony. This axis refers to the collaboration style that designers assume: they can make changes to the work being developed at the same time (a *synchronous* work style) or they can work at different times (engaging in an *asynchronous* work style) [7]. The higher the value in this axis, the more asynchronous is the work style.

Questions: do the team members change artifacts at the same time? Or do they make changes at different times? How frequently?

Distribution. This dimension describes whether work is being conducted at the same physical location or at geographically distant locations.

Questions: how far are the team members collaborating? Are they in the same building? Or are they in a different continent, or scattered through a country?

These dimensions can be effectively used to assess a given work style adopted by an interaction designer or a team. A single work style is plotted as a line (a point in the eight-dimensional space) whereas regions (or planes) represent sets of work styles.

Figure 2 also shows how this model was used to drive the development of a new user-centered tool for designing UI's (User Interfaces). CanonSketch [14, 10] is an UML-based tool that supports multiple levels of detail by providing the designer with three views: UML view of the UI and domain models, Canonical Abstract Prototype [12] and HTML concrete prototype (as the right side of Figure 2 exemplifies). The first two views are synchronized and the UML semantic model is used to support traceability. There is also a collaborative version of this tool in which designers can work at the same time on the model and at different places. However, support for distribution is still limited (for instance, there are no awareness mechanisms). Therefore, CanonSketch supports a region in our model, as illustrated in Figure 2.

In this way, the tool seamlessly supports designers while switching from high-level abstract views of the UI and low-level concrete realizations [14]. CanonSketch has been tested under a laboratorial setting and has lead to promising results. By contrast, a visual Interface Builder only supports a line in the work style model (the dashed line in Figure 2).

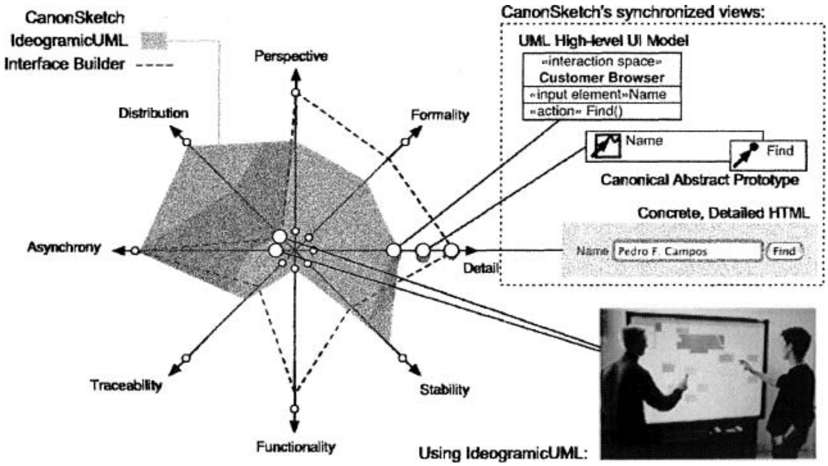


Fig. 52. Illustration of three different tools plotted under the Work Style Model.

IdeogramicUML, a gesture-based UML modeling tool, was also plotted in our model. It covers a considerable region, which suggests its adequacy to several styles of work during development tasks. IdeogramicUML only supports the syntax of the UML. However, it uses a sketch recognition language and can be effectively used in electronic whiteboards. There is also a distributed version with awareness mechanisms built in. Thus, one of the better supported transitions in work style provided by this tool is switching from synchronous, co-located development to asynchronous, distributed development.

11.1 3.2 A framework for studying Work Style support

In recent research results, we discovered that professional practitioners of interaction design engage into different work styles throughout their quotidian endeavors. We performed a survey which was distributed to professional interaction designers associations and mailing list, and collected 245 usable responses. This study, which is described in [15], had two main goals:

- Assess the practical aspects of the work style model: in particular if asking questions about work styles would be feasible and would lead to interesting findings;

- Find interesting patterns of tools' use and/or work style transitions among industrial designers.

Among other issues, we were interested in finding out which work style transitions did the practitioners considered more frequent and more difficult, in their everyday work practices. By frequent, we meant "how many times [the respondent] engages and transitions in those work styles", and by difficult we meant "how difficult [the respondent] finds to perform that transition". We confronted respondents with several concrete scenarios of work style transition and asked them to rate frequency and cost by selecting a value from a 7-point Likert scale, labeled with 1-low, 4-moderate and 7-high. Figure 3 shows the average rate for each transition.

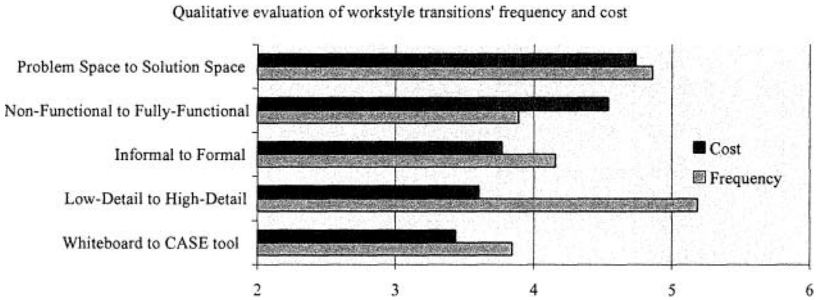


Fig. 53. Some transitions in work styles and their frequencies and cost.

Results showed that the most frequent transition was "moving from high-level descriptions of the user interface (sitemaps, navigation maps, etc.) to detailed screens (with concrete widgets, buttons, etc.)". This is a detail work style transition ("low-detail to high-detail". The second most frequent transition was also rated the most difficult one: "Moving from business rules, use cases and problem space concepts into final solution design, and back". This is a perspective work style transition ("problem space to solution space").

Based on the Technology Acceptance Model [6], current research literature [7], the Workstyle Model for UCD [10] and our survey's results [15], we designed an experimental framework aimed at studying the interaction designer's tools and work styles. Figure 4 summarizes the constructs in our framework, as well as the hypotheses we tested.

Perception-related variables operationalize the constructs of this framework. Four perception-based variables are measured, just like in the TAM:

- *Perceived Usefulness* (PU) is defined as the degree to which the user believes that using the tool will enhance his or her performance in designing interactive systems;

- *Perceived Ease Of Use* (PEOU) is defined as the degree to which the user believes that using the design tool will be free from effort;

- *Attitude toward using* (A) measures the feelings of favorableness toward using the tool;

- *Behavioral intention to use* (BI) measures the strength of a designer towards using the tool in the near future.

Work Style-related variables measure some aspects that come from our Work style model and from the transitions considered most difficult and frequent by professional interaction designers (according to our survey):

- *Perspective Transitions Frequency* (P) is defined as the rate of transitions from different perspective views, i.e. the frequency of transitioning from problem space concepts (use cases, task flows) to solution space (architecture, abstract prototype) and back;

- *Detail Transitions Frequency* (D) is defined as the rate of "drill-down" or "roll-up" between model elements, i.e. switching from high-detail views of an element to low-detail or the opposite;

- *Modifiability Rate* (M) is the rate of change made to any element of the artifact(s) being designed. This might include changing names, color, size, values or any other property of elements.

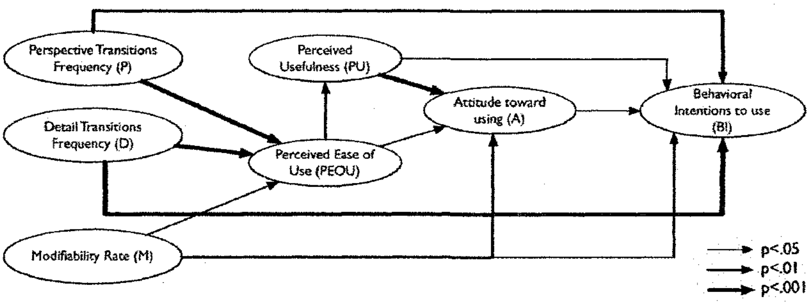


Fig. 54. Framework for combining Work Style analysis with Technology Acceptance analysis.

The framework was tested using logging tools for the quantitative measurements (P, D and M) and a survey that measured qualitative responses to PU, PEOU, A and BI. Figure 4 shows the revised theoretical model after regression analysis. The stroke of the arrows' thickness depicts the statistical strength of the relationships between factors. We observed how work style transitions have an influence on the tools' perception of usability and usefulness as well as behavioral intentions to use it.

If perspective and detail transitions are viewed by professional interaction designers as the most difficult (perspective) and frequent (detail) kind of transitions, and if our results show that these transitions' frequency has negative impacts on the tools' perceptions and intentions of use, then tool designers should find innovative ways to ease those transitions. The same happens with modifiability: the results suggest that the more modifications, the lower the positive feelings regarding the tool. Since we have showed that almost 80% of the time is spent modifying artifacts, effort should be targeted at easing this activity.

This framework of influences, together with our survey's instrument and Work Style model constitutes the core of our framework. We used it to design and evaluate a new set of design tools (CanonSketch and TaskSketch), aimed at improving the design experience and achieving more usable products.

12 4 Sketching and Designing Design Tools

We have been combining Work Style modeling with the well-established principles of Usage-Centered Design (U-CD) with the objective of designing and evaluating better design tools. In this section, we will present the interpretations and actual implementation of some of our most interesting designs.

4.1 Usage-Centered Models and Sketches

In this section, we describe some of the most interesting models, based on Usage-Centered design principles we applied during the process of designing new tools.

Usage-Centered Design is a methodology that tries to get the right design right from the start [9]. The twin foundations upon which U-CD is built are *process* and *principles*. U-CD defines an orderly and efficient *process* to transition from problem space to solution space. The process is guided by *principles* of what is considered good design, design that is most likely to allow the users accomplish their goals. Neither the U-CD process or principles guarantee one will design a good solution, but following both process and principles will improve the odds.

The essence of U-CD lies at three core models: the *user role* model, the *task* model and the interface *content* model. The difference between this and other design methodologies is relying on abstract, technology-independent models which prevent the designer of getting lost in the details of the design or in technology commitments.

We combined U-CD process and principles with our work style analysis framework for designing new tools, i.e. we apply U-CD to the design of U-CD tools. concerning *users*, we are interested solely in the roles they play in relation to a system, and we capture the prominent and noteworthy aspects of these associations in the shape of an abstract user role model. Figure 5 shows the user role model coupled with the most significant tasks each role is related to.

We define a "General-Modeling Role" which is sometimes specialized into a "User-Roles Modeling Role", a "Task-Cases Modeling Role", "Content Modeling Role", and a "UI-Realizing Modeling Role". Each of these roles is assumed by the user (the interaction designer using the tool) and each of them represents a different position in the work style space. In our tools' design and implementation we have tried to give support to a fluid transition between these any of these roles.

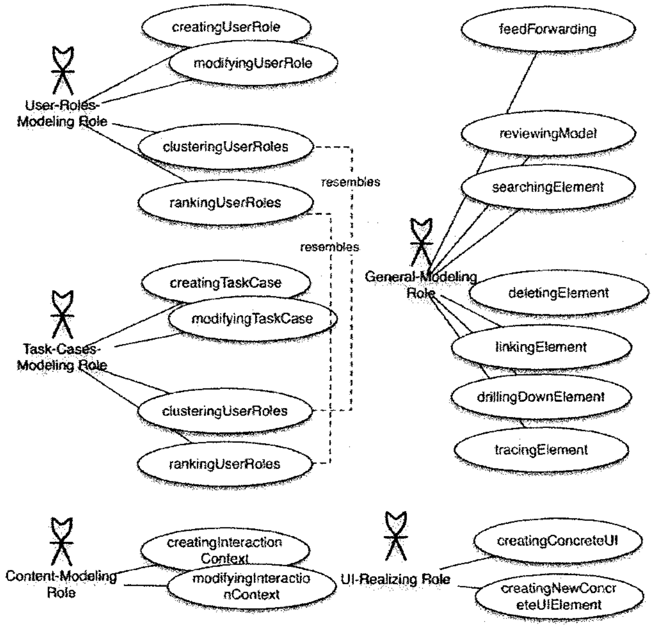


Fig. 55. User Role Model for a Usage-Centered Design Tool (full model omitted for brevity).

Another noteworthy aspect is related to the resemblance between some of the tasks. In empirical studies we discovered the designer is engaged into two main activities: creating (roughly 25%) and modifying (roughly 75%). Modifying a design also means reviewing it. This makes "Creating…", "Modifying…" and "Reviewing…" the central tasks in our model (although this is not explicit, we kept it in mind during the concrete, final design).

Figure 6 exemplifies how one proceeds to the task model. For a task model, we turn to task cases: these are use cases defined by abstract, generalized, technology-free descriptions [9, 12]. These can also give rise to interesting design ideas: as an example, consider the *linking element* task case. We declare that it is a system responsibility to adequately "show the designer available link types". As the user "chooses link type" (e.g. an aggregation, composition, etc.), the system should "provide feedback on valid sources". After choosing a valid source, the system should then "provide feedback on valid targets".

While there are many possible realizations of designs that support this task description, the truth is that it can be effectively realized in different platforms or even different modalities. Regarding different platforms, we only implemented it in the Mac OS X environment, but we are already working on a Windows design and implementation. Regarding different modalities, we are implementing it by using a pen and tablet system (using sketch recognition libraries) and by using voice input and output (simply using a speech recognition library and a microphone plus speakers).

linkingElement

User Intentions	System Responsibilities
	show available link types
choose link type	
	provide feedback on valid sources
choose source model element	
	provide feedback on valid targets
choose target model element	
	place link

drillingDownElement

User Intentions	System Responsibilities
choose model element	
request a drill down	
	provide detailed info
optionally request to edit	
	show editing tools
edit element	

tracingElement

User Intentions	System Responsibilities
choose model element	
request trace	
	show related models
choose model to trace	
	show related elements in the model

creatingConcreteUI

User Intentions	System Responsibilities
review task cases to support	
review content model	
choose abs. component to realize	
	show possible realizations
choose realization	
	place it

Fig. 56. Some Essential Task Cases for Design Tools.

The main point here is that these abstract, essential descriptions help developers of design tools to focus on what is really important to users (in this case, knowing which sources/targets are valid dynamically). It also helps to be technology independent because we avoid early design decisions that could prevent us from exploring different designs. In the next section, we will describe concrete sketches of our design tools. The sketches were made using Apple's Interface Builder, and we tried to give support to the tasks in our U-CD model as well as to smoothly support the most significant and important work style transitions.

4.2 CanonSketch and TaskSketch

CanonSketch [10, 14] and TaskSketch [16] are two closely related innovative modeling tools that try to address major weaknesses in current model-based tools. They have proved to be effective in promoting innovative visual and interaction designs that better support user performance.

CanonSketch (available at http://dme.uma.pt/canonsketch) is the first tool to support abstract user interface prototyping using canonical abstract components. Canonical abstract prototypes serve as an intermediary between task and object models on the one hand and working user interface prototypes on the other.

The tool enables rapid modeling and prototyping through three synchronized views at different levels of abstraction: UML class model, canonical abstract prototype, and functioning HTML prototype.

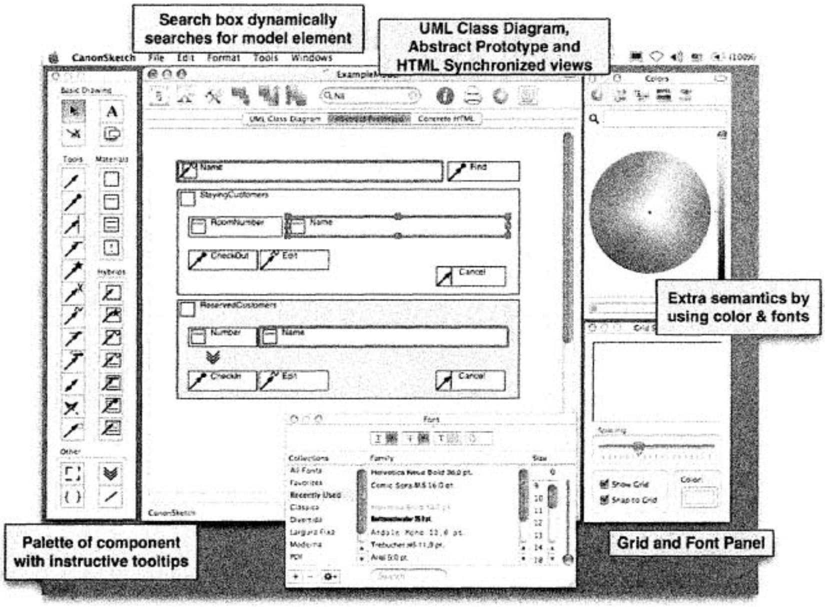


Fig. 57. CanonSketch illustrated.

Figure 7 illustrates a sketch of CanonSketch's main interface. The three synchronized views at different levels of abstraction try to support work style detail transitions. Since we showed that reviewing and searching a model significantly occupies the time of a designer, we provide a search box that highlights one or more model elements dynamically (i.e. as one types).

Extra semantics was also found to be important, because designers often extend the modeling language to convey extra meaning. This is done through using color and fonts (we are also implementing and testing a "star-rating" mechanism that helps designers classify the most urgent or important model elements).

TaskSketch (available online at http://dme.uma.pt/tasksketch) is an interactive requirements elicitation and modeling tool focused on linking and tracing use cases. It supports collaborative modeling by multiple stakeholders, including clients, marketing staff, and software engineers. It is unique in facilitating the development and exploration of the conceptual architecture based on use case narratives developed in essential form. It enables tracing the requirements of a system, in terms of user intentions and system responsibilities, to the conceptual architecture of that same system, making it easy to extract that architecture from task flows and to prioritize development of the most important classes.

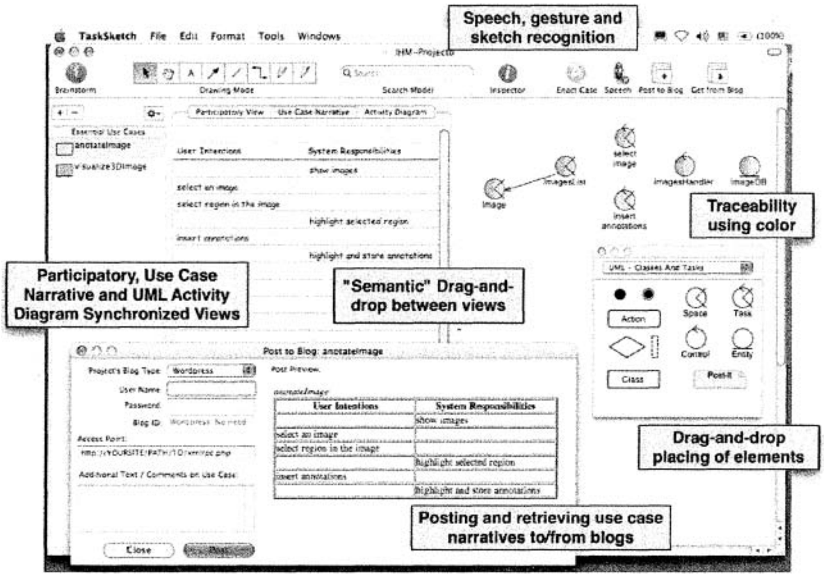


Fig. 58. TaskSketch illustrated.

Figure 8 illustrates some aspects of this tool. Traceability between related model elements is achieved by using color; the developer can look at the architectural view of the system and see which classes handle which use cases. This simple support to requirements traceability can be very powerful for, e.g., prioritizing development by deciding which classes are more urgent to implement. TaskSketch also supports what we call a "semantic drag-and-drop" between views. For example, if the designer drags a user intention from the use case narrative view and drops it into the architecture view, the tool creates the corresponding architectural element (in this case a task).

Following the same spirit of CanonSketch, this tool also provides the designer with three synchronized views for describing use cases and task steps. The Participatory view that uses post-it notes is good end-users and clients. The Task Case Narrative view is targeted at Usage-centered designers, and is nothing more than a digital version of index cards commonly used during this process. The UML Activity Diagram view is adequate to software engineers that practice sound, rigorous development methods such as the Rational Unified Process. Each user of TaskSketch can thus choose it's preferred work style and stick to it having the possibility of transitioning to another style whenever desirable (a software engineer working using the UML view can rapidly switch to the Participatory view to communicate models to a client or end-user).

Finally, we are also concentrating effort on exploring the possibilities offered by gesture recognition, mixing formal and informal notations and collaborative development using speech recognition and a shared display. This accounts as an effort to support collaboration styles and formality transitions.

13 5 Principles and Practice

In designing CanonSketch and TaskSketch, many issues were raised both from informal observation of the tools' usage and from formal evaluation studies in laboratorial settings. We will try to summarize a few of the most relevant issues, starting with "what worked well", then describing "what didn't work well".

5.1 What worked?

"Semantic" drag and drop. This idea came up as a means to support the Traceability and Perspective dimensions. Designers can drag elements from the activity diagram view (that describes task flows) and drop them in the architecture view, where they are translated to their semantic equivalent. This feature lowered the cognitive load designers were faced when switching views or perspectives and was very appreciated.

Blogs as use case repositories. We wanted to better support collaboration work styles, so we used blogs as common use case repositories fostering offline collaboration (different time, different places). Users can post comments, make changes using web browsers and simply browse the use case narratives, and then designers can choose one or more blog posts and retrieve them into the semantically-sound TaskSketch models. Figure 8 illustrates this mechanism.

Brainstorm Collaborative Environment. Also inline with supporting collaboration work styles, we wanted to give users a tool to work at the same time but at different places. The brainstorm collaborative environment, where users concurrently post and cluster task cases, classes or UI ideas, proved very attractive to users, probably because of its' Messenger-like interface or the dynamics of graphics.

Dynamic Search Box. The dynamic search box that smoothly highlights model elements as the user types proved very useful not only as a search means but also as a way to "filter" desired elements (sharing common properties from the designer's perspective).

5.2 What didn't work?

Tool palette as opposed to Drag-and-Drop palette. CanonSketch used a Tool Palette "a la" MS Paint, which users tended to dislike (although appreciating the tooltips). Following users' suggestions we implemented a drag-and-drop palette ("a la" MS Visual Studio) in the TaskSketch tool. Apparently, users found this approach more usable.

Lack of automatic alignment. Clearly these design tools need to improve or adopt automatic alignment mechanisms of some sort. As we have shown, high modifiability rates negatively influence one's attitude toward adopting a design tool, so it is imperative to add this concern when deploying tools for designers. Simply drawing the model elements on the screen, or having a grid layout like MS Visio won't do.

Use Cases manipulation view. Users didn't find comfortable using the "table-like" interface for creating and manipulating the use cases. This is probably related to the process itself: use cases are the central constructs from which everything else is modeled, but they need to be effectively clustered and manipulated.

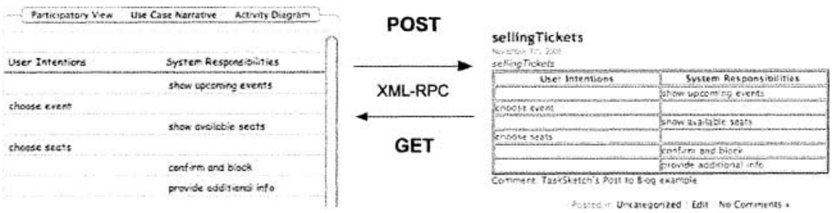


Fig. 59. The "posting to/getting from" blog mechanism illustrated.

5.3 Principles based on Work Style Analysis

Based on what we learned during the design and evaluation of both CanonSketch and TaskSketch during a 2-year period, we suggest - and in some cases we try to illustrate - a set of concrete design principles for tool developers.

Explorability. Design tools should make it easy to explore design alternatives. Basic undo-redo mechanisms are a poor man's approach to support this. Designers should not be penalized for trying out other models/solutions and not being able to go back or change ideas. What the Work Style model also suggests is that tools should invite creativity in early design stages and force focus and discipline later on.

Expressiveness. One of the major areas of weaknesses in current modeling tools is related to its' expressive power. A significant skill required to interaction designers is the ability to express and present their design ideas. Tools should provide helpful mechanisms to achieve this.

Guidance. Depending on the notation being used, the design tool should "just-enough" constrain the actions of the designer. Designers need efficient guidance also on the form of concrete design principles, not canned solutions.

Desirability. Models created with the tools should look engaging and attractive. The user interface should be stylish. Designers are more prone to adopt a design tool based on the visual design of the models created with that tool. A desirable and engaging user interface

In general, we believe that a tool supporting regions of our work style model will clearly aid the design and analysis tasks faced by developers of interactive products. Providing support for multiple levels of detail and formality enables designers to start with low-fidelity sketches of products, then gradually add the precision and detail needed to test a prototype. The flexibility achieved by such a tool will most likely contribute to the ease of learning of both tool and method, accommodating to the user's work style and usage impulses.

14 6 Conclusions

Work style modeling has proven a successful, practical method (when coupled to U-CD) of designing a new set of design tools. Our framework has been used not only to design but also to evaluate design sketches and analyze the work practices of interaction design, both at academia and industry.

There are many advantages that arise from work style modeling. Supporting work style transitions is a good way to increase the odds of sketching solutions that provide a tighter fit between everyday work practices and the system being developed. Our work style framework also informs design: thinking about the transitions between values in work style dimensions can give rise to new design ideas and serve as a means to validate design decisions.

Describing and modeling the styles of work users were engaged with has helped us design more usable UCD tools, such as CanonSketch and TaskSketch. Usability studies have shown better results and we believe work style modeling can be very useful both as an informal discussion tool and as a Human-Work Centered Design approach.

References

1. Ilvari, J. (1996). Why are CASE Tools Not Used? *Communications of the ACM*, 39:94-103.
2. Seffah, A. and Kline, R. (2002). Empirical Studies on Software Developers' Experiences. In WESS'02: Eighth IEEE Workshop on Empirical Studies of Software Maintenance. October 2nd, 2002. Montreal, Canada.
3. Clemmensen, T., Orngreen, R. and Pejtersen, A. M. (2005). Describing Users in Contexts - Perspectives on Human-Work Interaction Design. Workshop Proceedings of Interact'05, Rome, Italy.
4. Morris, M. G. and Dillon, A. (1997). How User Perceptions Influence Software Use. *IEEE Software*, 14:(4), 58–55.
5. Wright, P., Dearden, A. and Fields, R. (2000). Function Allocation: A perspective from studies of work practice. *International Journal of Human Computer Studies*, 52:(2), 335-356.
6. Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology. *MIS Quarterly* 13:(3).
7. Wu, J. and Graham, T. C. N. (2004). The Software Design Board: a Tool supporting Workstyle Transitions in Collaborative Software Design. In Proceedings of the EHCI / DSV-IS'2004, Hamburg, Germany.
8. Nunes, N. J., Cunha, J. F. (2001). WISDOM: Whitewater Interactive System Development with Object Models. In Mark van Harmelen (Editor), Object-oriented User Interface Design, Addison-Wesley, Object Technology Series, 2001.
9. Constantine, L. and Lockwood, L. (1999). Software for Use. A Practical Guide to the Models and Methods of Usage-Centered Design. Addison-Wesley, Reading, MA, pp. 194-195.
10. Campos, P. and Nunes, N. J. (2005). Galactic Dimensions: A Unifying Workstyle Model for User-Centered Design. In Lecture Notes in Computer Science, Volume 3585, pp. 158–169.
11. Landay, J. and Myers, B. (2001). Sketching Interfaces: Toward More Human Interface Design. *IEEE Computer*, pages 56-64.

12. Constantine, L. (2003). Canonical Abstract Prototypes for Abstract Visual and Interaction Design. In Jorge, J., Nunes, N. and Falcão e Cunha, J. (eds.), Proceedings of DSV-IS'2003 – 10th International Workshop on Design, Specification and Verification of Interactive Systems, LNCS - Lecture Notes in Computer Science. Springer-Verlag.
13. Seffah, A. and Metzker, E. (2004). The Obstacles and Myths of Usability and Software Engineering. Communications of the ACM, 47(12): 71-76.
14. Campos, P. and Nunes, N. J. (2004). CanonSketch: a User-Centered Tool for Canonical Abstract Prototyping. In Proceedings of the EHCI/DSV-IS'2004, Hamburg, Germany, 2004.
15. Campos, P. and Nunes, N. J. (2006). Tools of the Trade: the Practitioner's Tools and Work Styles. Submitted for Review.
16. Constantine, L. and Campos, P. (2005). CanonSketch and TaskSketch: Innovative Modeling Tools for Us-age-Centered Software Design. In Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA'05), Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, San Diego, CA, 162–163.

A simple design for a complex work domain – the role of sketches in the design of a Bachelor study's new folder structure for use by teachers, students and administrators

Torkil Clemmensen

Department of Informatics, Copenhagen Business School,
Howitzvej 60, DK-2000 Frederiksberg
Tc.inf@cbs.dk

Abstract. In this paper we explore the role of design sketches in Interaction design and work analysis in a case of designing a simple folder structure for e-learning software used to do course administration at a higher education study programme. The case presents a detailed description of how developers use different work analyses to collectively reflect upon and interpret design sketches of possible support of different user groups' interaction within their complex work, learning and life contexts. We conclude with what was learned from the case make recommendations how to conceptualize the process of reading design sketches using work analysis.

1. Introduction

Design sketches contribute both to the outcome of design and development processes, as well as to a greater understanding of the work itself. For the development of a science of interaction design it is therefore crucial to understand the role of design sketches in the interaction design process. In this paper we explore the role of design sketches for interaction design and work analysis and then suggest an approach that we call an 'interpretative approach to reading design sketches.' An interpretative approach to design sketches within the framework of Interaction design may have three steps: 1) sketch(es) of interaction design and the empirical work analysis of human life and work contexts need to be outlined and presented, 2) the design sketch and the work analysis are connected through separate analysis, for example, an intertextual analysis that consists of comparing work analysis reports with design sketches, and 3) additional theory is used to reflect on the readings of the design sketch(es).

Empirical analysis of work within an interaction design framework may contain different analyses. First, analysis of the organizational usefulness of the future design may include analysis of meeting agendas and resumes, consultant reports, organizational content templates and policies, interviews with key individuals in the

---

Please use the following format when citing this chapter:

Clemmensen, T., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 221–240.

organization, and other methods [21, 22, 25]. Second, analysis of the individual usefulness of the future interaction design may include the creation of conceptual models, i.e., explicit ideas about how the future users should interact with the new design. Third, analysis of the different kind of work procedures that the new design will support may include task analysis. Fourth, analysis of how users in the future may use the new design can be envisioned in, for example, scenarios. Fifth, analysis of who the future users are may include the construction of personas, i.e., fictive users that are representative for a target group of users of the new design. Finally, the analysis of the user's mental interaction with the new design may include think aloud tests. In contrast to the many techniques available for work analysis, the design sketch technique itself does not receive any systematic treatment (see for example [7] or it is characterized as simple pen and paper drawings of icons, dialog boxes, etc. [21].

The connection of design sketches with work analysis may happen through separate analysis. We suggest an approach that relies heavily on end-user involvement through discussion groups when comparing the specific work analyses with the different design sketches. Such analyses may include comparison to organizational standards for such designs or semantic mapping analysis, such as: Are the words, concepts, etc, that are used in the design sketch taken from the work analysis? Does the design sketch convey the moods and feelings that the work analysis suggested? Does the design sketch illustrate how a task is supported? It may also include analysis of how the design as sketched should be maintained and how it will be compared to competitors' choices of similar designs. Many other analyses are possible. What may be critical, however, is going beyond the textbook approach of interaction design to sketches[9] in order to develop a set of separate user-oriented analyses that connects work analysis and sketching.

Reflecting on the readings of the design sketch(es) involves the use of additional theory beyond Interaction Design Theory. First of all, the theory behind empirical work analysis, for example [16, 22, 25], and behind 'cognition in the wild' [14] is relevant to understanding what work analysis is and how to develop how work analysis can be used to say something about the design sketch. Second, we need the theory of design sketching [10, 19, 20, 27] to appreciate what the design sketch contributes to interaction design, including theory about how we by reading the design sketches also learn more about the work analysis.

[9] The interaction design textbook of Preece, J., Rogers, Y. and Sharp, H. Interaction Design: Beyond Human-Computer Interaction. John Wiley & Sons, 2002. has one entry on sketches. This entry discusses the designer's anxiety related to showing his or her inadequate drawing skills.

The rest of the paper explores the interpretative approach to reading design sketches by analysis and discussion of a case of the development of a new Bachelor's study folder structure for use by teachers, students and administrators;

Figure 1 illustrates the new folder structure at the end of the case period. (The figure is in Danish. Each entry in the folder structure represents a student class).

Figure 1. A simple design for a complex work domain: The folder structure of a Bachelor's study programme.

We first provide some background for our approach to reading design sketches. Second, the research methodology is outlined as an 'action research' oriented case study. Third, we present the case and analyze: how the developer group approached work analysis through organizational analysis, task analysis, scenario development and usability testing; how these analyses were applied in the discussions and interpretation of sketches and prototypes that were designed and used during the development of the folder structure; and how the developer group's use of design sketches reflected possibilities for supporting different user groups' interaction within their various work, learning and life contexts. We conclude with lessons learned from the case, and recommend how to conceptualize the process of reading design sketches.

1.1 Background: design sketching and work analysis

Interaction design is presented in textbooks as an approach consisting of conceptual models, scenarios, HTA task analysis, persona, think-aloud evaluation and other user centered techniques [7, 21]. These techniques may be seen by more engineering oriented designers as 'work analysis' techniques and not directly related to design [5]. However, textbook interaction design also includes prototypes, storyboards, sketches, etc. Sketches, i.e., freehand drawings or low fidelity prototypes, have been studied by researchers for their role in design and have been found to stimulate reflection, particularly in the early stages of design [20].

When going from analysis to design, from conceptual models to physical design, interaction design relies heavily on iterative testing of prototypes with users of the future product [21]. A large number of techniques for user requirement elicitation and user tests are available for use in interaction design [16, 21, 25]. In many of these techniques, communication between stakeholders about user requirements is supported by the use of prototypes, mock-ups, etc. However, not much is said about the use of freehand drawing sketches (sometimes called low fidelity prototypes) when going from analysis to design. For example, it is not well defined how to use organizational analysis, task analysis or scenarios in combination with one or more design sketches.

Having a focus on how to use work analysis to read design sketches is a different approach from design cognition that asks questions such as: "Do abilities of the designer, like general intelligence, visual abilities regarding imagery and perception, and creativity, influence the usefulness and quality of sketching?"[1, 11]. The focus is also different from research in artifical intelligence support to design, which tries to build computational tools for design evaluation that may present critiques in the forms of text, diagrammatic annotation and 3D model to the designers' sketches [19]. It is also different from studies of design practice that may try to describe how designers during design imagine their users [13]. Using work analysis to read design sketches is a different approach from the approach used in studies of how to use ethnographic field methods in participatory design [12], though it can be seen to be closely related [2, 23]. Instead, we see the use of work analysis to read design sketches as an interpretative approach.

2. Research methodology

In the sections below we present a case study [8]. of the development of a new Bachelor's study folder structure for use by teachers, students and administrators in a Computer Science and Business Administration programme. This development process is studied within a one year period at the researcher's own university (Copenhagen Business School[10]).. The sources of information included background reports, emails and notes from meetings, videos of test situations and, in particular, design sketches from the development group. Further, the author of this paper played a central role as chair or the developer group in the case. The context of the case was a decision taken by the university's top management that ordered the study board to

[10] Copenhagen Business School has around 14,000 students and an annual intake of around 1,000 exchange students. With this number of students as well as around 400 full-time researchers and around 500 administrative employees, CBS is the one of the 3 largest business schools in Northern Europe.

stop using their own in-house developed course administration system and begin using the university's standard course administration system. From this followed the need for designing the folder structure of the new system's in a way that accommodated and in some cases changed the course administration process known to the users of the old system.

This case illustrates an interpretative approach to reading design sketches using work analyses as input. The analysis of the data is focused on the use of design sketches. We provide a description of different types of work analysis done by the developer group, how these analyses were used in the discussions of the sketches, and how the sketches were annotated and interpreted and reflected upon during the process. We unfold the context of the case by presenting a chronology of meetings in the developer group with a particular focus on the meetings in which design sketches played the major role. Finally, we report on the lessons learned from the case study.

3. Case Study Analysis and Results

In October 2004 the members of the study board received an email saying:

> *"The deans and the university administration have decided that the platform Sitescape now is mandatory for all courses and all students at the university. Therefore you at your study program have to begin using this platform no later than autumn semester 2005."*

The email marked the end of a year-long political discussion in the study board of the value of retaining the old in-house developed course administration system called DIVE at the study programme. It was also the beginning of the transition from the old to the new system that is described in this case. The analysis and the results of the case study are presented below.

3.1 The work analysis

The study board's response to the direct order from the head of the university to abandon DIVE in favor of the university's current standard system called Sitescape was to appoint a developer group. The developer group had as its goal to make the transition from the old system to the new system bearable and workable for all user groups. It consisted of representatives from teachers, study administration, students and experts from the university learning lab, with the author of this paper as the chairman. This developer group began its work half a year before the new system had to start working and finished seven months after that date, i.e., the period during which the developer group functioned was a little more than a year. During that period the developer group held a number of meetings (see Table 5 below) to analyze the course administration work performed by different user groups and to develop the new folder structure. The work analyses are described in the following sections.

Organizational analysis

The organizational analysis of the course administration process and the old system was done by the developer group on the basis of archival data and knowledge from within the group. A consultant report of needs and requirements for a new system had been authored by a former student at the study programme. This student had worked extensively on this topic during his studies, had been a member of the previous as well as the current study board, and was seen as one with some insight into the work procedures surrounding the course administration. The consultant report on course administration had three main points: 1) the need for one-way communication from the study administration to the students and teachers, 2) a continuous need for re-organizing the structure of the material used in courses (for example, teachers needed to establish new shared locations for material to all students as new topics were introduced in teaching), and 3) the need for a 'branding' of the study through design of the system, both internally towards students, teachers and secretariat, and externally towards potential applications for the study. Furthermore, the developer group became aware of the existence of a content template for 'study-zones' (?the folder structures system of the new system) authored by experts from the university's learning lab and authorized by the central study administration at the university. This template made it clear that the developer group should leave out tasks related to study administration and focus on the tasks related to course administration.

Task analysis

The task analysis of the course administration work was done as two different analyses. Two sub-developer-groups, a student and a teacher/administration group, did each a task analysis. As seen in Figure 2 and Figure 3, the student group did it as a 'use case' analysis [17]. This task analysis was confirmed by a task analysis done by another student group that also showed that checking for new teaching material and sharing documents in relation to teaching were two main tasks of course administration for students.

Use Case: Find teaching material

1. Find year
2. Find subject
3. Find module
4. Find teaching plan
5. Find teaching session

Figure 2. Student sub-developer-group's task analysis of 'find teaching material'

Use Case: Knowledge sharing in a student group

1. Establish group
   a. Find group members
2. Install permissions
3. Agree on rules for cooperation
4. Make structures within the student workgroup
5. Upload documents

Figure 3. Student sub-developer-group's task analysis of 'knowledge sharing in a student group'

The teacher sub group did a task analysis that showed that upload and sharing of documents was the most frequently performed task by teachers. Typically, they had to upload their presentation slides to make them available to students. Apart from uploading of slides, teaching plans, assignments, etc., no other task was required of teachers in relation to course administration. However, the teachers' tasks were not independent of the pedagogical strategy that was used in the study programme; many teachers reported informally of the experiences with managing discussions, online supervision of assignment work and other interactive tasks. Furthermore, the task analysis showed that the setting up of new system users was a task that required getting access to other systems at the university, selecting users, naming groups, and more, and therefore was a task that should be done mainly by the study administrators. Teachers were also responsible for archiving data, i.e., establishing the memory of the study programme.

Scenarios and personas

On the basis of the use cases, the developer group developed several scenarios [3] with associated personas [18]. In one scenario, Hjalte and Christian were two bachelor students who wanted to establish a study group, to share documents and to find material from the teacher. In another scenario Peter, a senior teacher with a lot of experience, wanted to set up a forum for his teaching, upload a large number of files, and monitor the changes, for example, if students had commented or asked questions regarding the material uploaded by the teacher. Pther kinds of scenarios were important too; for example, one of the experts from the university's learning lab presented four generic scenarios that illustrated different visions of how to teach with the support of the system: lecturing, class teaching/preparation, class teaching/problem solving and team assignment/project. Taken together, these scenarios with their included personas suggested ideal future users and ideal future situations of use of the new Sitescape system in course administration. These scenarios were presented at an appropriate time (at a meeting in August) for all teachers and students at the study programme.

DEVELOPER GROUP MEETINGS

January 2005

Establishing goals (design of folder structures) and resources (limited to workgroup)

February

Organizational analysis and task analysis: Establishing user subgroups: students, administration, teachers. Presenting expertise from central unit and experience from other study-programmes. Distributing consultant reports on old versus new system and manuals and papers about the new system. Giving workgroup members access to new system

March

Sketches from user subgroups (student, teacher/administrator)

April 2005

Hi-fi prototypes from expert designer

May 2005

Folder structures implemented on new system

June 2005

Scenarios and Think aloud tests of implemented folder structure

August 2005

Involving all teachers and students in analysis of the use of the new system as a pedagogical tool

September -December

New system is operative.

January 2006

Evaluation of the first semester with the new system

The old system information established as an archive at new system

3.2 Using the work analysis

At the same time that the developer group indulged in the above presented work analysis activities, the developer group came up with several design sketches in the form of low fidelity prototypes. Sketches illustrating the points and arguments of the sub developer group were presented at meetings in the developer group (see Table 5 for a list of the meetings in the developer group). These meetings took place between the large meetings. During the meetings the sketches were annotated with points taken from discussions in the group. In this way, the sketches were connected to the work analyses through discussions in the developer group and used as input for the next round of analysis and sketching. The annotated design sketches are presented below.

Table 5. Meetings in the DIVE (old system) - Sitescape (new system) developer group

3.3 The design sketches

During the developer group's period of existence, several design sketches in the form of low fidelity prototypes were made. Each sub-developer-group provided sketches illustrating the points and arguments of the group. The student sub-developer-group did two sketches: one of the principles of organizing the folder structure and one illustrating an example of an unfolded structure.

The students' sketch of the principle folder structure that was created is shown in Figure 4; the annotations on the the left side of the figure relate to the classes, those in the middle refer to the courses, and those on the right comment on the student groups; the row at the bottom of the figure explains the symbols used in the figure. The students presented their sketch as a 'long-life', transparent and easy-to-use folder structure for students as well as teachers. They argued for a structure with two levels and a third level consisting of folders only. They requested that the third level (the right-most column in Figure 4) be tailored to each individual course.

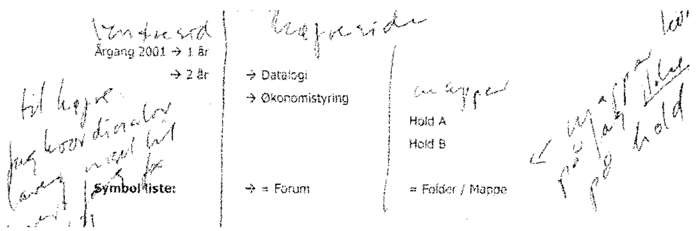
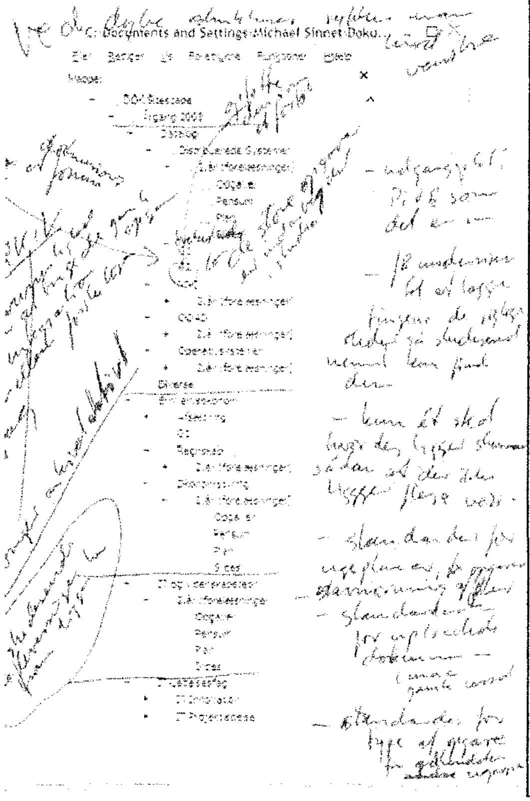


Figure 4. The students' sketch of the principle in the new folder structure

The annotations on the principle sketch in Figure 4 also reflect the developer group's discussion that showed the students wanting their teachers to be primarily responsible for tailoring the structure on the lower levels to the need of the students. A translation of the annotation on the left is: "The course coordinator makes some information for each course...", which suggests that the students wanted the coordinator to take responsibility for the content. The annotation on the right says: "folders should be on courses, not on student groups" which again moved the responsibility for the information from the student group to the teachers. This built-in division of work between students and teachers was made even clearer by the sketch produced by one of the students of the unfolded structure (see Figure 5).



Annotations translated:

- New system Sitescape look and feel like old system DIVE
- Teachers to upload documents in the correct locations so that students could find them quickly and easily
- Timetables to be at one place only, not scattered around
- Standards for week plans, e.g. ,assignments
- Standards for naming
- Standards for uploading of documents (mac-files, word, ppt, pdf, etc.)
- "Big" assignments embedded in the study folder structure
- Students allowed to deliver their assignments electronically in the appropriate folder

Figure 5. The students' sketch of the unfolded structure

The students' sketch of the unfolded structure in

Figure 5 shows a deep unfolded structure with each student class at the highest level, the courses at second level, and inside these, the folders for each course. Although the figure is in Danish, the handwritten annotations from the meeting discussion of the sketch are translated into English. From the annotations it is clear that several points were not explicitly stated by the students but had to be elicited during group discussion (as seen on the right part of

Figure 5). As these points illustrate, the students wanted the teachers to think seriously about improving their use of the folder structure; e.g., teachers should obey standards for naming, document format, etc.

The left side of the teacher sub-developer-group's sketch of the folder structure in Figure 6 shows the structure for each semester, the right side of the sketch folders referring to lectures, notes, messages and exercises classes, and the right bottom relating to general information.

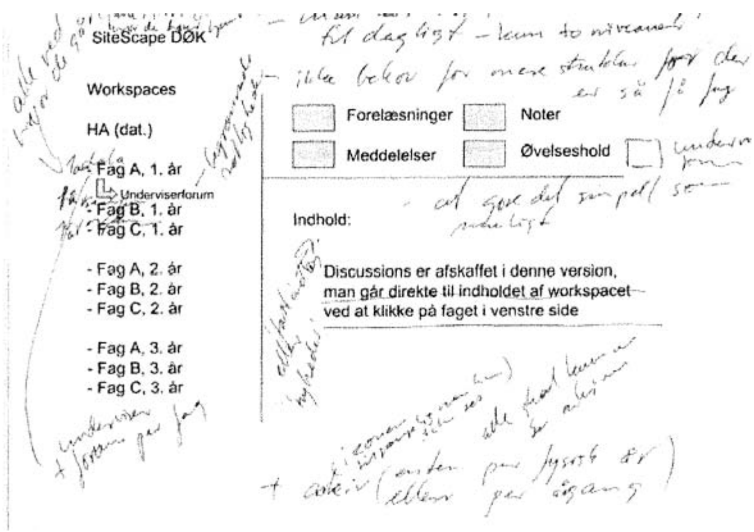


Figure 6. The administration/teacher sub-developer-group's sketch.

The written annotations on the sketch (in Danish) reflect the outcome of the developer group discussions of the teachers' sketch: a) for daily teacher use there is only a need for two levels of structure since there are so few courses, b) the teachers want it to be as simple as possible due to their heavy workload often being distributed across several studies, and c) an additional forum for teachers only should be added to the structure. Furthermore, the developer group considered the sketch - in its content organization and layout - as much dominated by the IT support individual. She had a clear grasp of how the new system appeared in several of the other study programmes at the university and wanted something similar. She wrote in an email:

> *...in my opinion it has no relevance to develop a whole new proposal for a zone construction [the folder structure], I rather think it pays of to look at this cand.merc. zone [a competing study programme's folder structure] ...then it must be up to those who make the final solution to take into account all requirements..."*

It came out strongly from the presentation and discussion of the administrator/teachers sub-developer-group's design sketch that the teacher representatives, together with the study administration representatives, saw a need for a simple structure that complied with organizational standards. This was because teachers and administrators had to work on several study programmes and did not want to specify interfaces for any of these study programmes.

The sub-developer-groups' sketches, i.e., the students' and the teachers' design sketches, and the discussion of these in the developer group led the developer group to conclude that the learning lab expert should develop new high-fidelity prototypes illustrating what different solutions would look like in the real system.

The expert's sketches

The high-fidelity prototypes developed by the learning lab expert illustrated different solutions, i.e., a hierarchical folder structure vs. a flat folder structure. A flat hierarchical combined prototype was also produced, but did not make any impact in the working group and will not be presented here.

The prototypes presented in Figure 7 and Figure 8 had similar top level structures (a teacher forum and a student forum at the top, and student developer groups at the bottom), but different lower level structures (hierarchical vs. flat). The prototypes were implemented in the existing Sitescape system as dummy pages with access codes only to developer group members, in order to illustrate how the folder structure would appear on the screen, including font, text organizing, pictures, links, and integration of other web sites.

The annotations request:

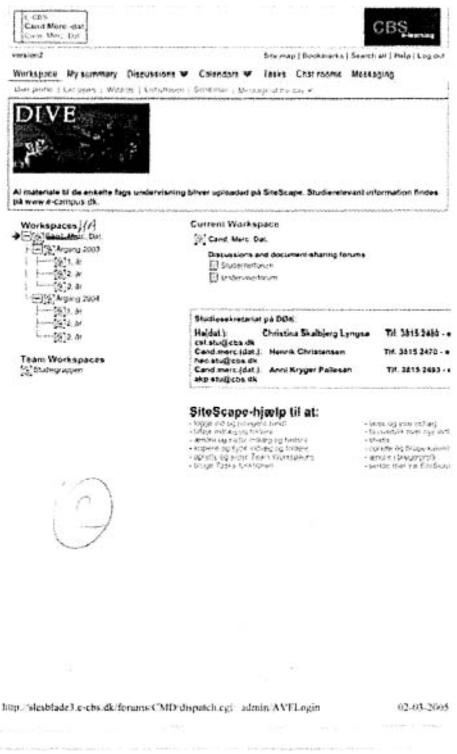
- Semesters indicated at the top-most level
- Teaching forums for all of the bachelor students
- Teaching forums for 2nd year teachers only, for sharing inputs to a special cross-disciplinary student assignment
- Teaching forums only for 3rd year teachers only, for sharing thoughts about supervision of final student projects
- A special folder for announcing courses for those who go on to the Master 's programme

Figure 7. Prototype 1. The expert's sketch of a flat folder structure

The group discussion of the prototypes focused on advantages and disadvantages of each prototype; each participant in the developer group was required to access the prototypes and test the prototype for three bad things and three good things, and be ready to report these at the meeting in May (see Table 5).

In the high fidelity prototype 1 (see Figure 7 which is in Danish with translations at the right) the expert designer illustrated one version of a flat folder structure, with all the courses listed at the highest level. The expert designer's arguments for prototype 1, apart from the flat structure in itself, were that advantages included fewer clicks, only current courses on the screen, and the possibility for several discussions under each course. The disadvantages which the expert perceived were that it would take more work to archive data after each semester and that the user had to go to another place in order to find both old courses from the current year and material from the previous years.

However, the expert designer of the high fidelity prototype 1 had mixed the Master's programme with the Bachelor's programme, evident from the attempt to correct the wrong study programme name in the annotations made during group discussion (marked 'A' in Figure 7). Also, the annotations on the expert's sketch reflect how during discussion of the sketch the group became aware of the need for several forums for teachers, not only one.



The annotations say that there should be a semester name at the top-most level, instead of the (wrong) study programme name

Figure 8. Prototype 2. The expert's sketch of a hierarchical folder structure.

In the high fidelity prototype 2 (see Figure 8) the expert designer illustrated a hierarchical folder structure that was an exact copy of the old DIVE system folder structure. The expert designer's arguments for the advantages of prototype 2 included that it was easy to go the courses from one's own class, and in general other classes were easily accessible; the disadvantages included too many clicks, a deep structure, folders within folders made it harder to attain an overview than folders in Discussions (a standard folder given by the system). Additionally, the teacher forum needed to be outside courses and not within courses (due to a folders inheriting access rights form Discussions standard folder). As the lack of annotations in prototype 2 reflects, the developer group did accept these arguments without further discussion.

At the May meeting (see table 1) the group chose between the prototypes and recommended version 2 to the study board on the basis of its resemblance to the old system and the good features that were attributed to the overview and accessibility.

Evaluation of the prototype

A think aloud testing of the chosen prototype 2 was done using scenarios as scripts for role play. Representatives from each user group (students and teachers11, see Figure 9 and

Figure 10) were asked to describe a typical scenario and then to perform this scenario within the new system while thinking aloud. This was recorded on videos, which were then searched for usability problems and benefits.

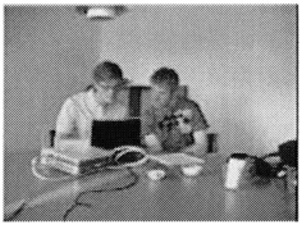
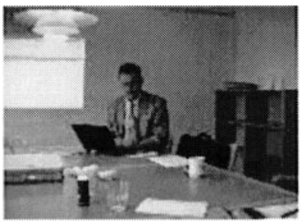


Figure 9. The students doing think aloud test of prototype 2.



[11] The user groups that did think aloud testing did not include the administrators, because they had already at a special meeting in June got supervision from the learning lab expert to use the system to do their task of registering new courses.

Figure 10. The teachers doing think aloud test of prototype 2.

The results of the think aloud tests were contradictory; on the one hand, the system including the new folder structure was ineffective and unsatisfactory to both students and teachers; one the other hand, the folder structure seemed easy to learn and easy to re-learn, quite secure and also compatible with other study programmes' use of the system. In other words, the think aloud tests in both test user groups showed mixed benefits for students' and teachers' use of the new system.

At a later occasion (in August) the think aloud videos were shown at an open meeting for all those associated with the study programme. This audience of teachers and study board members and experts from the university's learning lab expressed general satisfaction with the new system, confirming that the system was easy to learn to use. However, a consultant with e-learning expertise, who had experience of the system from other study programmes at the university, but now saw saw prototype 2 for the first time, clearly indicated that the developer group had made the wrong choice, that is, they should have chosen prototype 1, which other study programme had done. This left the question of the quality of the new folder structure somewhat open for a time.

After implementation evaluation

Prototype 2 was actually implemented as the study programme's new course administration system, as seen in
Figure 11. After having operated the new system for a whole semester, the developer group at a meeting (in December) planned the evaluation of the folder structure and discussed feedback from different user groups. As chair of the developer group, this researcher sometimes received emails from teachers. For example, a cc was received of one teacher writing to another teacher:

"I can see you have uploaded some eclipse.jar, but I still cant se what it is, because I still don't have access to the teacher forum on 1st year!"

Another teacher wrote to this researcher directly:

" I have been fully satisfied with Sitescape which I have only used as a place for uploading documents to the students."

It appeared that those of us in the developer group had received only part of the story - not the full story from all perspectives - of the transition from the old to the new system with a new folder structure for use by teachers, students and administrators.

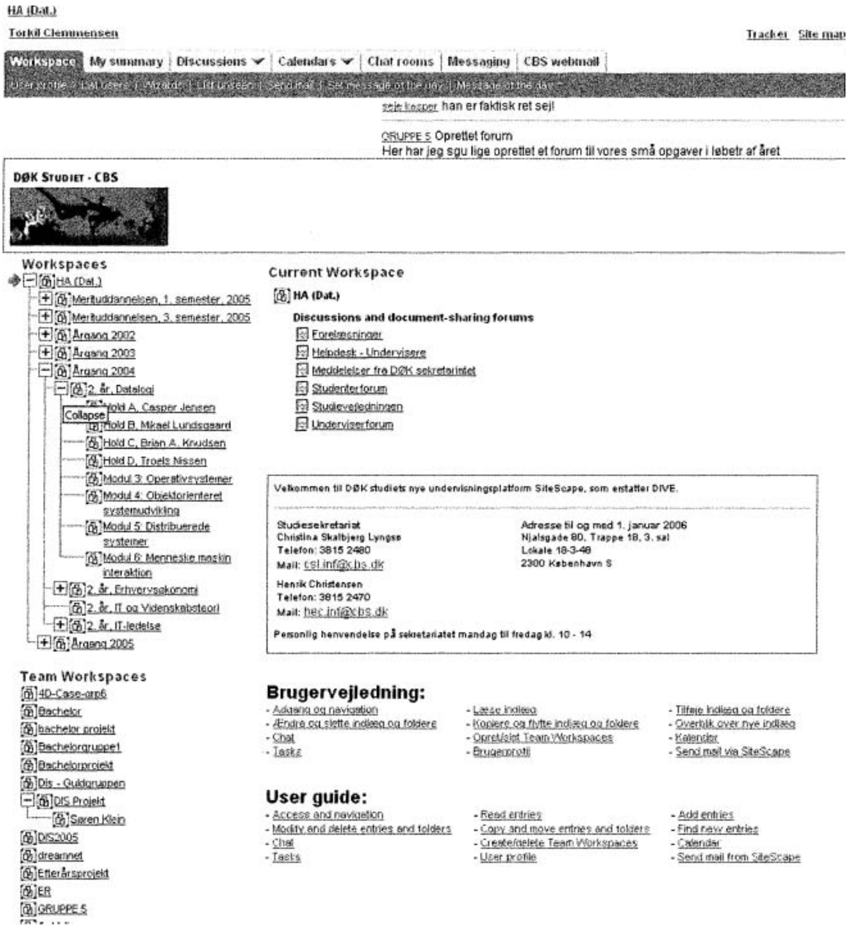


Figure 11. The implemented version of prototype 2 folder structure.

3. Discussion

The case described above of moving from an old to a new system for course administration illustrates how interpretative processes are at play on several levels of the organization in order to connect work analysis and design sketches.

Design sketches are tools in work analysis

One lesson learned from the case was that the collective discussion of the design sketches revealed much about the course administration work itself. From a work analysis perspective, a theoretical framework that may illuminate understanding of the use of design sketches in the case presented here is distributed cognition, a theory that purports that both individual and collective cognition can be and generally is distributed across time and space [6, 14], for example, in a flight cockpit [15] or in a courtroom or a medical practice [9]. Having this focus makes it interesting to analyze how information is propagated across different representations during a specific period of time. In this perspective, information about how to do course administration was propagated across the developer group's design sketches. The design sketches were used as working memory registers that enabled the group to share the immediate thoughts. However, it is not always easy for everybody to share immediate thoughts; thus, this process had to be supported by formal think aloud testing of the prototypes. The developer groups' collective use of the design sketches as memory registers also made the design thinking more robust and relevant than if one member of the developer group sequentially stated his or her requirements of the new design. In this way, the developer group became part of the course administration in a time of difficulties – at one occasion it wasn't clear whether the developer group did course administration or development of course administration. The lesson learned from the case is that within a distributed cognition perspective, design sketches are tools in the ordinary human activity that the development activities support? (in this case the course administration). This means that a design sketch could be evaluated not as the outcome of work analysis but as a contribution to work analysis.

Design Sketch Ownership

One obvious critique of the research presented here is that the sketches that are analyzed above are not really design sketches, but are PowerPoint low fidelity prototypes or the equivalent. According to the dominant view of design sketches (Orngreen, this issue), a design sketch is a hand drawing that conceptualizes an idea and which has not been taken too far towards something that can be presented and perceived as a solution. In contrast to this view, the case study presented here illustrates that solution spaces can and will be explored by the use of *any* kind of drawing including power-point low fidelity prototypes and other computer drawings by end-users and other stakeholders. Somebody owns the design sketch, that is, each sketch is an expression of someone's perspectives and feelings towards the work to be supported. From a design sketch perspective, the use of different representations, sketches, low and high fidelity prototypes for communicating with other people and hearing their views. In particular, the use of sketches with annotations of what is good, bad and how they meet requirements is recommended by practitioners[12].

In the case presented here, the use of different representations, sketches, low and high fidelity prototypes was necessary to address the various levels of organizational learning about (and at the same time performing) the teaching and study administration using the new folder structure raised by the organizational analysis initiated at the beginning of the developer groups meetings. Each user group needed at least one sketch, as reflected in different sketches from the students and the

[12] See e.g. http://www.kurtz.ws/teaching/l542-v2/assgn.htm, retrieved Monday, December 19, 2005 or http://www.sapdesignguild.org/editions/edition1/miniapp_design.asp, retrieved Monday, December 19, 2005.

teachers. Furthermore, each institution needed separate sketches of the new structure. The university's learning lab expert provided prototypes that were based on the sketches and ideas discussed in the developer group but which clearly twisted the ideas in new directions. This cognitive and social answer to the use of sketches is different from the creativity enhancing role that current theory of design sketching identifies [10, 19, 20, 27]. It seems its focus is on who presents the sketch more than an understanding of design sketches as cultural and emotional carriers. [26].

The design sketch as mediating artifact

It appears from the case that analyzing work is one process, drawing design sketches is another, and connecting the sketch and work analysis is a third process. The connection of design sketches with work analysis may thus be understood as an entirely separate analysis (in addition to work analysis and design sketching), for example as a form of analogical reasoning about novel design concepts, or as a kind of *intertextual analysis of design sketches and work analysis reports.*

In analogical reasoning about novel design concepts, it is worthwhile recognizing that there is a difference between design sketches (including low fidelity prototypes) and high fidelity prototypes: It may be easier to explain design concepts by use of design sketches than by use of high fidelity prototypes,\ because design sketches facilitate analogies between domains, while high fidelity prototypes (examplars) facilitate reasoning within domains [4]. Thus, in our case example design sketches may facilitate analogical reasoning between the old system and the new system.

In an intertextual analysis, the role of the developer group meetings in the reading of design sketches may be that of providing collective interpretations of the sketches by inclusion of end-users, a process whereby the complexity of the design becomes apparent. In other words, what is designed is both the artifact (the folder structure), the work procedures (the interaction with the folder structure) and the human beings (what is means to be a student, teacher and administrator) – a perspective that in recent years has been made popular in information system research as actor network theory [24]. In this case, the lesson learned through the intertextual perspective is that work analysis results are compared with design sketches, i.e., evaluating sketches (with or without end-users of the new design being present) includes comparison to organizational standards for such designs, to task analysis, to scenarios and to other forms of texts. It may also include analysis of how the design as sketched should be maintained and how it could be compared to a competing design choices.

4. Conclusion

The interpretative approach to understanding the role of design sketches in interaction design and work analysis which is presented in this paper suggests that the interpretations take place during meetings by oral commenting and by written annotations on the sketches. The comments and annotations, i.e., the interpretations, may be understood as a kind of intertextual analysis that compares and connects work analysis reports and design sketches. Furthermore, the presented case makes it clear that many sketches are needed for even a simple design when the work domain is complex.

5. Acknowledgement

Gratitude is shown to the anonymous members of the working group at the study programme for their enthusiastic input and the design suggestions made during the one year period. The case study was made possible due to support by the head of the study board, Jacob Nørbjerg. Appreciation is also due to Karl Kautz, Dept. of Informatics, CBS for commenting upon early drafts of this paper.

6. References

1. Akin, Ö. Variants in Design Cognition. in Eastman, C., Newstetter, W. and McCracken, M. eds. *DESIGN KNOWING AND LEARNING: COGNITION IN DESIGN EDUCATION*, Elsevier, 2001, 301.
2. Blomberg, J., Suchman, L. and Trigg, R.H. Reflections on a Work-Oriented Design Project. *Human-Computer Interaction, 11* (3). 237-265.
3. Carroll, J.M. Five reasons for scenario-based design. *Interacting with Computers, 13* (1). 43-60.
4. Christensen, B.T. and Schunn, C.D. The relationship between analogical distance to analogical function and pre-inventive structure: The case of engineering design. *Memory & Cognition.*
5. Clemmensen, T. and Nørbjerg, J. Separation in Theory - Coordination in Practice. *Software Process Improvement and Practice, 8*. 99-110.
6. Cole, M. and Engestrøm, Y. A cultural-historical approach to distributed cognition. in Salomon, G. ed. *Distributed cognitions – Psychological and educational considerations*, Cambridge University Press, 1993.
7. Cooper, A. and Reimann, R.M. *About Face 2.0: The Essentials of Interaction Design*. Wiley, 2003.
8. Creswell, J.W. *Qualitative inquiry and research design - choosing among five traditions*. SAGE, London, 1998.
9. Engestrøm, Y. Interactive expertise - Studies in distributed working intelligence. *Research Bulletin, University of Helsinki, Department of Education, 83*.
10. Fallman, D., Why Research-oriented Design Isn't Design-oriented Research. in *Nordes: Nordic Design Research Conference*, (Copenhagen, Denmark, (2005)).
11. Hamel, R., Psychology and Design Research. in *Design Research in the Netherlands*, (1995).
12. Harper, R.H.R. The Organisation in Ethnography- A Discussion of Ethnographic Fieldwork Programs in CSCW. *Computer Supported Cooperative Work, 9*. 239–264.

13. Hasdogan, G. The role of user models in product design for assessment of user needs. *Design Studies, 17.* 19-33.
14. Hutchins, E. *Cognition in the wild.* MIT Press, 1995.
15. Hutchins, E. How a Cockpit Remembers Its Speeds. *Cognitive Science, 19* (3). 265-288.
16. Kirwan, B. and Ainsworth, L.K. (eds.). *A guide to task analysis.* Taylor & Francis, 1992.
17. Mathiassen, L., Munk-Madsen, A., Nielsen, P.A. and Stage, J. *Object Oriented Analysis and Design.* Marko, Aalborg, 2000.
18. Nielsen, L., From user to character. in *Designing Interactive Systems*, (London, 2002).
19. Oh, Y., Gross, M.D. and Do, E.Y.-L., Critiquing Freehand Sketching - A Computational Tool for Design Evaluation. in *Visual and Spatial Reasoning in Design [VR'04]*, (MIT, Cambridge, USA, 2004).
20. Oh, Y., Yi-Luen Do, E. and Gross, M.D., Intelligent Critiquing of Design Sketches. in *AAAI Fall Symposium - Making Pen-Based Interaction Intelligent and Natural*, (2004).
21. Preece, J., Rogers, Y. and Sharp, H. *Interaction Design: Beyond Human-Computer Interaction.* John Wiley & Sons, 2002.
22. Rasmussen, J., Pejtersen, A.M. and Schmidt, K. Taxonomy for Cognitive Work analysis, Risø National Laboratory, 1990.
23. Siegel, D. and Dray, S. Avoiding the next schism: etnography and usability. *Interactions, march + april 2005.*
24. Tatnall, A. and Gilding, A., Actor-Network Theory and Information Systems Research. in *10th Australasian Conference on Information Systems, 1999*, (1999).
25. Wilson, J.R. and Corlett, E.N. (eds.). *Evaluation of human work - a practical ergonomics methodology.* Taylor & Francis, London, 1992.
26. Yammiyavar, P.G. Emotion as a semantic construct in product design, IIS, Bangalore, 1999, 367.
27. Yi-Luen Do, E., Gross, M.D. and Zimring, C., Drawing and Design Intentions -- An Investigation of Freehand Drawing Conventions in Design. in *Design Thinking Research Symposium*, (Cambridge MA, 1999).

Design as Dialogue – a New Design Framework

Arminda Lopes

a.g.lopes@leedsmet.ac.uk
Leeds Metropolitan University
Innovation North
Faculty of Information & Technology

Abstract. In this paper, we examine some traditional ways of designing and we present some guidelines for a different perspective on design: design understood as a process of dialogue and explicit knowledge sharing. There is no standard definition of design it will vary according to perspective. Traditionally, the term design referred to industrial design. This argument is based on Walker (1989) consideration: "*(As far as many scholars are concerned, design is industrial design.)*" Industrial design concentrates on the product's functionality and its appearance as an object. Design as dialogue requires an emphasis on the process of communication between multidisciplinary teams. We will provide a framework based on dialogue which will improve the process of design associated with the development of flexible design objects.

Introduction

Design is a complex process. There isn't a single definition of design – it can be understood from different perspectives. Designers rarely work alone. They cooperate with others, interacting with groups of people who have different ways of looking at the design problem and the design solution. They bring different knowledge to the design project, and differing viewpoints, expectations and ambitions.

Dialogue is also a complex process without a unique definition. Ellinor (1995) argues "*dialogue is about what we value and how we define it. It is about discovering what our true values are, about looking beyond the superficial and automatic answers to our questions. Dialogue is about expanding our capacity for attention, awareness and learning with and from each other. It is about exploring the frontiers of what it means to be human, in relationship to each other and our world*".

The goal of this paper is to present some guidelines based on the perspective of design as dialogue. We consider some definitions of design and observations from the literature; we consider three specific usages of the word design: designed object, design process and design discipline. We describe a way of thinking about design: design as dialogue, explaining in what ways this is the case and the benefits of this perspective: We initiate a discussion of a framework based on this perspective, which would help to simplify the design area, and to facilitate the dialogue process. We conclude by highlighting the key research issues and outlining our work in progress in this area.

Please use the following format when citing this chapter:

Lopes, A., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 241–250.

What is Design?

Determining what is meant by design is not straightforward. The word "design" has different meanings according to the designer's community, the context in which the word is used and the speaker's and listener's perspectives: whether economic, cultural, social, or aesthetic.

Most of our conceptions of design come from industrialisation. This claim is based on Bayley (1982) who considers "*the origin of design with the advent of industrialization and mass production methods.*" In general, design is the act of working out the form of something, for example by making a sketch, outline or plan.

Dictionary definitions also focus on modelling and planning. For example, according to 21st Century Dictionary, design (noun) is "*a plan, drawing or model showing how something is to be made. Design (verb): to develop or prepare a plan, drawing or model of something before it is built or made.*"

The Concise Oxford English Dictionary defines design (noun) as "*a plan or drawing produced to show the look and function or workings of something before it is build or made.* Design (verb): *decide upon the look and functioning of (something), especially by making a detailed drawing of it; do or plan (something) with a specific purpose in mind.*"

Those definitions overlap, the difference being that the concepts of aesthetic and functionality included in the second one.

However, the etymology of design goes back to the Latin *de+signare* meaning distinguishing it by a sign, giving it significance, designating its relation to other things, owners and users.

By analogy, the roots of the word dialogue come from the Greek words *dia* and *logos. Dia* means "*through*"; *logos* translate to "*word*", or meaning. Isaacs (1999) pointed out that "*In essence, a dialogue is a flow of meaning.*"

We really need to state some definitions about design in order to understand why being design a fertile and challenging subject it is complex and also it need to a new framework, the dialogue perspective which will allow establishing new ways of understanding design.

14.1 Observations about Design

In the literature relating to this area we can find some observations that have been made about design and which are important to understand why design is an activity fundamental to human action and why design as dialogue is a useful framework.

Design history is important because narrations of design have a feedback effect in the way it can influence the guidelines of the future of design. Doordan (1995) pointed out that design history, like other areas of historical inquiry, "*is constantly reconfiguring itself, reformulating its subject matter, and redefining its methods in order to contribute in a vital way to the discussion of contemporary issues and opportunities.*" His main focus is that design is a process of change itself. Designers will communicate the changes by creating it, contributing to innovation and using design as a social and cultural process. Walker (1989) considers design as "*a fertile and challenging subject for the historian because it occurs at a point of intersection or mediation between different spheres, which is between art and industry, creativity* *and commerce, manufacturers and consumers*". For him, design can also be considered as an aspect of human culture.

Buchanan (1998) refers the cultural, technological, communicative aspects of design. He argues that design "*is a liberal art of technological culture, concerned with the conception and planning of all of the instances of the artificial or human-made world: signs and images, physical objects, activities and services, and systems or environments*". He considers that there are three great expressions of design thinking in the twentieth century: engineering, marketing and the forms of graphic and industrial design.

Multidisciplinary is an important characteristic in design as in dialogue. Matthews (1998) said that "*a simplest engineering design involves several different engineering disciplines. Design is sometimes intangible, repetitive and iterative; it contains lot of the world's complexity*". Pugh (1996) design "*it is an activity that integrates the bodies of knowledge present in the arts and sciences*".

Julier (2000) considers design as "*a focus of leisure and consumption; it has become a source of public entertainment and serves as an indicator of economic performance, cultural regeneration and social-well-being.*"

Preece (2002) referring the design process, pointed out that the aim of interaction design is "*bringing usability into the design process, which means developing interactive products that are easy, effective and enjoyable to use, from the user's perspective*".

Findeli considers that design can have different definitions depending on if it is considered to be an idea, knowledge, a project, a process, a product, or even a way-of-being.

These observations are reflecting different aspects of design: change, culture, communication, multidisciplinary, complexity and usability. All these observations are important aspects related with dialogue as well as design which we will state in this paper.

A Way of Thinking about Design: Design as Dialogue

Design can be understood from different perspectives, as stated earlier. It involves communication between people through a dialogue.

Gadamer (1979) cited by Smith (2001) describes conversation as: "*[It] is a process of two people understanding each other. Thus it is a characteristic of every true conversation that each opens himself to the other person, truly accepts his point of view as worthy of consideration and gets inside the other to such an extent that he understands not a particular individual, but what he says.*"

Design as a communication process is a dialogue or a conversation (we use both, dialogue and conversation, referring to the same meaning) between designer and other participant in the process.

14.2 What is Design as Dialogue?

Design as dialogue is a communication process between all the participants and elements of design. All designs tell something, through text, image, symbols, or styles. It is a conversation between designer and user. According to Isaacs (1999) "*dialogue is a conversation in which people think together in relationship. Thinking together implies that you no longer take your own position as final*".

Designers communicate directly with their users through the appropriate placement of visible cues, hints, and agreements. They use a narrative (a story, an interpretation of some aspects of the designed object).

Dialogue allows us to share meaning, look for understanding, make connections and interpretations, and inquire into assumptions. It invites us to discover. Design is also about all of this.

Dialogue can develop common values and allow participants to express their own interests. It expects that participants will grow in understanding and may decide to act together with common goals. Through dialogue, participants can question and revaluate their assumptions.

14.3 Elements of Dialogue (Communication, Meaning, Interpretation, Understanding)

Design as dialogue can only be understood in the context of communication, interpretation, meaning, understanding, learning, cooperation... We will state some of these elements which will allow analysing the dialogue between designers.

In a very broad sense, **communication** means to give and exchange information. And in this sense, normally we refer to communication among people or among non-human animals. People communicate with words and with pictures, and also communicate in many different languages.

Communication is also concerned with communication across cultures. Communication from a cultural perspective includes individuals, meanings (as theoretical meanings, values, norms, and ethics) and vehicles (as language and behaviour). Hall (1997) pointed out that "*to understand culture we need to explore how meaning is produced through the signifying practices of language.*"

Through dialogue, searching out **meaning**, we become more critical because using language we are not just expressing ourselves alone, but with others. Language and discourse are not just means of expression but they make possible the interaction between the participants. In dialogue, when one person says something, the other person does not in general respond with exactly the same meaning as that seen by the first person. This difference in meaning allows participants to see something new and to integrate multiple perspectives and shared meaning. According to Crystal (1997) meaning "*is a basic notion used in language study in two main ways: first determining the signification of a message (...) secondly, meaning is used as a way of analysing the structure of language...*"

Isaacs (1999) considers that "*there are dimensions in our internal world that do not exist in the external in the same way, but which are nevertheless quite important. For instance, our internal world is characterized by* ***interpretation*** *that is our ability to make sense of what we see and hear and experience.*"

Usually, we believe the ideas we are expressing are the same ideas we initially formed in our own mind. But how our expression is received depends on who is receiving it and the way he interprets it.

An object has an initial impact from its appearance, touch, and feel. Then at a performance level, the focus is on use, on experience with the object. Experience has many facets: function, performance, usability and aesthetics. Then there is the interpretation level in the identification of the product with places, feelings, and people.

Dialogue is a way of coming to an **understanding** and a way of relating to participants. When participants listen to each other and relate, they learn new perspectives, reflect on their views, and develop mutual understanding. It emphasizes listening, learning and the development of shared understanding. In design, understanding who the users are, what tasks they will be performing, how much experience they have, what problems they foresee, what expectations they have for the object will contribute to the quality of the design. These aspects of the design process will be the subject for the dialogue between designers.

How Design is a Dialogue Process

The word design is used in different ways. The specific usages we will discuss are **designed object**, **design process** and **design discipline**.

When we refer to designed objects the association is frequently with industrial design, a field where designers are able to convey messages, and even propose new ways of living, through the creation of functional objects.

Designed objects focuses on the concept of form and function and creating objects, furniture and tools that serve their purposes and are at the same time aesthetically pleasing.

The design *process* on the other hand, as Mathews (1998) considers, consists of three established stages: conceptual design; embodiment design; detailed engineering design. In general, a design process contains the following basic key steps: design brief; statement of intent; product design specification; concept design; detail design; manufacturing and testing; sales.

By *design discipline* we mean the field of endeavour which encompasses many design activities including (but not limited to) planning, urban design, architecture, interior design, product design, graphic design, arts, engineer, performance, music. Every product crafted by humans is the result of design.

Designed Objects as Dialogue: Objects communicate – a well designed object communicates with people by many different messages. A designer can create an object that is articulate; its form announces its function and it actually conveys ideas to those who interact with it.

When first confronted with an object, a user will understand the first idea that the designer intended to convey, and the design will relate further ideas with meaning.

In an object design, there are three components, according to Norman (2004) "usability, aesthetics, and practicality" which interact to convey information about the user's pleasure using the object. Jordan (2000) considers that "*once people had become used to having appropriate functionality, they wanted products that were easy to use (...) Having become used to usable products (...) products that people relate to*". So, people have relationships with objects and they can dialogue with themselves about the objects. This is possible using Bohm's (1996) observation about dialogue: "*a dialogue can be among any number of people, not just two. Even one person can have a sense of dialogue within himself, if the spirit of the dialogue is present.*"

Design Process as Dialogue: Dialogue enables understanding the process of design and the communication between participants. According to Bohm (1996) the process of dialogue is a *"process of 'awakening', it entails a free flow of meaning among all the participants"*.

Dialogue begins with an idea as design begins. The key components include the sender, the idea, the evidence that supports that idea, and the sender target. Once the idea has taken some reasonable form, it can be shared.

Design is the process by which the needs of the customer are transformed into a product satisfying these needs. The design process contains the discussion of existing products – people's opinion about products – prototyping – people's feedback, team's conversations, and dialogue with stakeholders. It contains a variety of questions and answers: what is the practical function of the design? What part does appearance play in the design's function? What materials are suitable for the design? What construction methods are appropriate to the design? What are the likely social and environmental effects of the design?

Design includes multidisciplinary teams participating in discussions in the design process.

Design Discipline as Dialogue: The way to interact within this multidisciplinary field is by dialogue. Dialogue can be a strategy that offers us some clues in the situation of diversity. Dialogue is a learning discipline about people and about objects.

The design discipline is concerned with the social, cultural, philosophical study of design. It draws upon approaches from Design History to Cultural Studies.

Design discipline allows the analysis of historical and cultural codes and also to deal with new design challenges. Design by definition allows understanding past and future developments in the field. Since Pre-History, for example, people communicate through pictograms, another mark was the Egyptian civilizations that codify design elements. Historical understanding of an object's significance and value reside in a wide range of influences and ideas, a complex social and cultural mix, a long communication process. Referring to industrial design and understanding it involves an analysis of modern consumption and interpretation of our everyday life which is full of signs and elements of communication.

Dialogue is to explore design as it is. History permits us to understand why design is dialogue (there was always a communication process within it – verbal and visual, for example).

14.4 Benefits of Thinking of Design as Dialogue

Thinking about design as dialogue will give us some advantages through a new framework.

Dialogue enables teams to reach higher levels of performance. Higher levels of performance are the result of learning processes and interrelationships between people. The practice of dialogue contributes to the design of objects and systems that clarify all the design process through listening, hearing and participation. Planned actions and interactions are often difficult to predict with clarity and confidence. Dialogue helps us to find connection and meaning within the design process.

Dialogue invites discovery as design does. It develops common values and allows participants to express their own interests and ideas. In dialogue, participants can question and re-evaluate their assumptions. Through this process, people are learning to work together to improve usability and aesthetic in designed objects.

Thinking about design as dialogue also underlines some probable solutions to design complexity.

Framework to Simplify Design Complexity

The framework based on dialogue will improve the process of design associated with the development of flexible design objects. Essentially, flexible design expands choice, on what, when, where and how design teams design. It will be a reference where each participant, designer and probably user, because a designer is a user, has a role in the changing process of design, solving complex problems giving simple solutions. The framework tries to answer some questions, for example: who are the participants in the design process? How might each dialogue element be relevant in the design/dialogue process? How might each design element (line, shape, texture, space, size colour and value) be relevant in the design/dialogue process? What are the elements contained in the interaction process?

14.4.1 Facilitation of Design Process

Dialogue builds learning from thinking together as a design team which means that thinking together builds conceptions that lead to shared understanding. Dialogue facilitates a view of participants as both observers and the observed. Dialogue requires an absence of hierarchy which means dialogue can only occur when design team sees each other as colleagues in a mutual search for deeper insight. Generally, in a design process, there are leaders in the process which could be considerable different with this approach. Dialogue as Isaacs (1999) said, presents "*the opportunity to reflect and to be heard, and to reflect further on what it means to be heard.*" This argument underlines the importance of dialogue on design, through talking together and thinking together without hierarchies.

14.4.2 Multidisciplinary Dialogue

Multidisciplinary dialogue can be found in the design teams. Design teams are constituted by different people, with different backgrounds and different perspectives. To understand the behaviour of the group and the designed object it is necessary to recognise cultural and language differences between the participants.

Cushner (1996) considers that Culture consists of interrelated components of material artefacts, social and behavioural patterns and mental products. Culture is something that is made by human beings rather than something that occurs in nature. He defines two kinds of culture: "*Objective culture – refers to the visible, tangible aspects of culture, and includes such things as the artefacts people make, the food they eat, the clothing they wear, and the names they give to things; and Subjective culture – refers the invisible, less tangible aspects of a group of people which means people's values, attitudes, norms of behaviour and adopted roles.*"

Kroeber (1952) refers to culture being produced by the past actions of a group and its members. The members of a culture system share a set of ideas, and especially values. These are transmitted particularly from one generation to another by symbols.

Kuhnt (2002) considers that culture influences the attitude towards the usability of globally used software products. It can be approached according to the dependence on user's beliefs about usability cultural specific variables or according to the dependence on their cultural background. Sometimes, users may focus either on aspects concerning effectiveness, efficiency or satisfaction when using a product, which rarely happens. The International Standards Organization (ISO) defines usability as: "*the effectiveness, efficiency or satisfaction with which specified users achieve specified goals in particular environments.*" (ISO DIS 9241-11).

The multidisciplinary dialogue can also be found in language. Language is more than a functional means of communication within a culture. Human language reflects how individuals think and interact. If we deal with the interaction process between users and interactive products it requires taking into account a number of independent factors including context of use, type of task, and kind of user. Language can be used with different meaning as Isaacs (1999) pointed out. " *The first is the voice and language of meaning, the second is the language and voice of feelings and aesthetics, the third is the language and voice of power, the power of our actions.* " Dialogue includes all three of these voices: meaning, aesthetics, and power. There is a need to understand which language is spoken at each time to engaging in dialogue.

Problems to be Addressed

Although "Design as Dialogue" seems straightforward it brings some problems which need solutions.

Understanding multidisciplinary design dialogue is not simple because there is complexity in design, complexity in dialogue and complexity in dealing with teams of people in design who have multidisciplinary background.

Specifying context is an important process because context helps to first understand how objects within a universe of discourse or domain discourse, i.e., the set of terms used in a specific discourse, relate to each other to form a coherent whole.

Even so, we consider that the work in progress, specifically the observation and analysis of some case studies, will bring more understanding of the whole the process.

We believe that dialogue is the pathway to interpret and understand the design process contributing to providing guidelines to help the designer community to build designed objects and to review the design process in a simple way.

Conclusion

Design is complex but can be understood in terms of dialogue. To validate this argument we need to have a deeper understanding of the dialogue process as well as understanding multidisciplinary design dialogue between teams of people with different backgrounds.

The work in progress aims to gain this understanding providing a framework which could be usable by multidisciplinary design groups to ensure good working practices and to achieve the best and the most comprehensive design for the target community.

The expectations are that through dialogue, designers will be able to reach beyond any differences and disagreements towards a better understanding of how to improve design.

Future Work

The work in progress is the study of interdisciplinary art and technology teams designing interactive artworks. Teams have different constituencies, for example researchers from art, design, IT, computer science, engineering, architecture, cultural studies, and media studies. Each team's main goal is to look at human-computer interaction (HCI) theory, design and application of interactive technologies. The interactive artwork will allow the analysis of dialogue as well as design process of the interdisciplinary teams.

Analysing the process, communication tools, language, and values will derive guidelines for the support of such processes.

The methods used to collect data for this research will be centred on a qualitative study, a combination of research methods will be used to collect the data, including documentation, case study, interviews and observation.

Discourse analysis will be used to examine modes of speech used in dialogue, used as social narratives to explain the events. All the design process is video recorded, accompanied by textual descriptions; attending to the conversation formulations produced by designers in their dialogue together; observation of designers talking, listening to their conversations and describing their formulations; analysis of the work plans; observation of all the activities which take place and the relationship between activities and designers.

Acknowledgments

I would like to thanks to supervisors Tony Bryant and Janet Finlay for comments on earlier drafts of this paper and to Leonardo Network for the opportunity they gave to work with them. This work is part of a PhD project partly funded by PRODEP III - Medida 05 / Acção 03 "Formação Avançada de Docentes no Ensino Superior" Código Projecto: 05.03/C/00195.012/03.

References

1. Bayley, S.: Art and Industry, London, Boilerhouse Project, 1982
2. Bohm, David: On Dialogue, Routledge, London, 1996

3. Buchanan, R.: "Banzi's dilemma: design in contemporary culture", Design Issues, 1998
4. Concise Oxford English Dictionary, Oxford University Press, 2000
5. Crystal, David: The Cambridge Encyclopaedia of Language, 1997
6. Cushner et al.,: Intercultural Interactions – A practical Guide, Cross-Cultural Research and Methodology, Volume9, Sage Publications, UK, 1996.
7. Ellinor, Linda and Glenna Gerard: Dialogue: Rediscover the Transforming Power of Conversation 1995 Published by John Wiley and Sons Inc, USA.
8. Hall, S.: Representation: Cultural Representation and Signifying Practices, London and Thousand Oaks: Sage, 1997
9. Isaacs, William: Dialogue and the Art of Thinking Together: A Pioneering Approach to Communicating in Business and in Life. Random House, 1999
10. Jordan: Designing Pleasurable Products London Taylor and Francis, 2000
11. Julier, Guy: The Culture of Design, Sage Publications, London, 2000
12. Kroeber and Kluckhohn, C.: Culture: A critical review of concept and definitions, Cambridge, MA: Harvard University Press, 1952.
13. Kuhnt, Thomas Vohringer: The Influence of Culture on Usability, Masters Thesis, Freie Universitat, Berlin, 2002
14. Matthews, Clifford: Case Studies in Engineering Design, Arnold, London, 1998
15. Norman, Donald: Emotional Design: Why We Love (Or Hate) Everyday Things, Published by Basic Books, NY, 2004
16. Preece, Jenny: Interaction Design, John Wiley and Sons Inc, USA. 2002
17. Price, Stuart: Communication Studies, Addison Wesley Longman Limited 1996
18. Smith, M. K.: 'Dialogue and conversation', The Encyclopaedia of Informal Education, 2001, www.infed.org/bibio/b-dialog.htm
19. Walker John: Design History and the History of Design, Pluto Press, London, 1989
20. 21st Century Dictionary, Chambers, Harraps Publishers, Ltd

Printed in the USA