ISSN 0105-8517

A Conceptual Toolbox for Designing CSCW Applications

Susanne Bødker
Ellen Christiansen
Manfred Thuring

DAIMI PB - 489
December 1994

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF AARHUS
Ny Munkegade, Bldg. 540
DK-8000 Aarhus C, Denmark

A CONCEPTUAL TOOLBOX FOR DESIGNING CSCW APPLICATIONS

Susanne Bødker
Dept. of Computer Science
Aarhus University
Ny Munkegade
DK-8000 Aarhus C, DENMARK
Tel: +45-89423256
E-mail: bodker@daimi.aau.dk

Ellen Christiansen
Dept. of Computer Science
Aarhus University
Ny Munkegade
DK-8000 Aarhus C, DENMARK
(Present address:
Dept. of Communication
Aalborg University
Langagervej 8, box 159
DK-9100 Aalborg, DENMARK)
Tel: +45-98158522 x 7119
E-mail: ellen@hum.auc.dk

Manfred Thüring
empirica
Communications and Technology
Research
Oxfordstrasse 2
D-53111 Bonn, GERMANY
(Present address: BIFOA,
Universität zu Köln,
Universitätsstrasse 45,
D-50931 Köln, GERMANY).
Tel: +49-221-47603-11
E-mail: thuering@bifoa.uni-koeln.de

Abstract:

This paper presents a conceptual toolbox, developed to support the design of CSCW applications in a large Esprit project, EuroCODE. Here, several groups of designers work to investigate computer support for cooperative work in large use organizations, at the same time as they work to develop an open development platform for CSCW applications. The conceptual toolbox has been developed to support communication in and among these design groups, between designers and users and in future use of the open development platform.

Rejecting the idea that one may design from a framework *describing* CSCW, the toolbox aims to support *design by doing* and help bridging between work with users, technical design, and insights gained from theoretical and empirical CSCW research.

This is done through the construction and use of various kinds of scenarios. These scenarios, created by designers in cooperation with users, are seen as important means of communication between groups of users and designers and among groups of designers. Scenario construction is guided by empirical concerns on the one hand, i.e. actual investigatory and design work with users, and theoretical concerns on the other through checklists and prototypical examples. To help its users produce scenarios on their own, the prototypical examples are meant to trigger "good ideas" with respect to work-oriented as well as technical solutions.

Going through an example, the paper illustrates how scenarios can serve as boundary objects between groups in a major research and development project like EuroCODE.

Résumé:

Cet article présente une boite à outils conceptuelle. Cette boite à outils a été développée pour soutenir la conception d'applications "*CSCW*" (travail coopératif assisté par ordinateur) dans un large project Esprit, EuroCODE.

Dans le cas présent, plusieurs groupes de concepteurs travaillent à explorer l'assistance par ordinateur pour le travail coopératif au sein de larges organisations d'utilisateurs. En même temps, ces groupes développent une plate-forme ouverte pour développer des applications *CSCW*. La boite à outils conceptuelle a été développée pour assister la communication au sein de chaque groupe et entre différents groupes, entre concepteurs et utilisateurs, et pour les utilisations futures de la plate-forme ouverte de développement.

Rejetant l'idée que l'on peut concevoir le *CSCW* à partir d'un cadre de travail purement *descriptif*, l'idée générale derrière cette boite à outils est d'assister la *conception expérimentale* et d'aider à établir un pont entre le travail des utilisateurs, la conception pratique, et la littérature théorique et empirique à travers la construction et l'utilisation de plusieurs sortes de scénarios. Ces scénarios, créés par les concepteurs en coopération avec les utilisateurs, sont vus comme des moyens de communication importants entre les groupes d'utilisateurs et les concepteurs, et entre les groupes de concepteurs. La construction d'un scénario est guidée par les impératifs empiriques d'une part (i.e. le travail conceptuel et exploratoire avec les utilisateurs) et les impératifs théoriques d'autre part, au travers de listes de contrôle (checklists) et d'exemples prototypiques. Pour aider les utilisateurs à produire leurs propre scénarios, les examples prototypiques visent à provoquer de "bonnes idées" relatives au travail du projet et à ses solutions techniques.

Au travers d'un exemple, cet article illustre comment les scénarios peuvent servir comme délimiteurs entre groupes dans un project majeur de recherche et de développement comme EuroCODE.

Keywords: Cooperation in design, CSCW framework, scenarios

INTRODUCTION

This paper presents a certain design idea coming out of the Esprit III project, EuroCODE[1]. This design idea is not a technical solution in the traditional sense, but a conceptual toolbox for supporting the design of CSCW applications in EuroCODE. EuroCODE as such aims at building a development environment for CSCW applications, consisting of an open CSCW shell, three demonstrator systems, and a CSCW framework, or as we have come to call it a conceptual toolbox [Bødker et al. 93b]. What this means is that we aim at offering support for developers at a technical and at a conceptual level. The shell provides a number of building blocks which can be used to develop a CSCW application from scratch or to enhance one of the demonstrator systems with additional functionality, and the intention of the conceptual toolbox has been to support this process. The demonstrators utilize the shell and toolbox in building three prototype applications within the application domain (two supporting cooperation in the supervision of the construction of the Great Belt Bridge [Grønbæk et al. 93], and one for long-distance radiology at a hospital, Rikshospitalet [Holmes 94]).

As framework constructors we have faced the situation within EuroCODE that a) the shell and demonstrator design is distributed, geographically as well as with respect to the organization of work, b) too many persons take part for all to be involved in the investigation of, and cooperation with the actual users at Great Belt and Rikshospitalet, c) too many different competencies are needed for everybody to take an interest in empirical studies, or in theoretical CSCW. Although a

[1] The EuroCODE name stands for **Euro**pean **C**SCW **O**pen **D**evelopment **E**nvironment.

lot of our previous work has assumed that such a distribution is *not* the best way to proceed, we find it, nevertheless, a fact of life, in this project as well as in other large design projects.

Given this situation, the EuroCODE framework has as such to serve a range of practical purposes:

- to transfer knowledge within the project,
- to make the knowledge gained in recent years' research in participatory design and CSCW available and working in technical design sessions, and to direct these experiences towards change,
- to make the shell and demonstrators available to people outside the project,
- to systematize and theoretically ground the empirical experiences, especially scenario construction,
- to focus the development,
- to help provoke thoughts and ideas, in scenario construction as well as in evaluation of these from a technical as well as from a work-oriented point of view.

Experiences from previous projects indicated to us that a method, in the traditional sense of a universal recipe, is not a feasible path. E.g. in Utopia, traditional requirement specification was substituted partly by experimental, participatory design based on mock-ups [Bødker et al 87, Ehn 88]. [Kyng in press] promotes scenarios as a way of supporting joint idea-generation, based on experiences from EuroCoop/EuroCODE, and [Bowers & Pycock 94] describe how design ideas come out of co-construction processes between users and designers. By borrowing the cooperative approach from Utopia and combining it with theoretical knowledge in an operational form, we seek to acknowledge the value of theory-driven design without ignoring the situatedness of use. Due to the distributed nature of our project, however, our conceptual toolbox needed to move beyond this.

From [Kyng in press], we derive five practical reasons for situational description in design: they are needed to grasp user experience, to get real world reference, to avoid failure due to the blindness of designers, to provide material for mock-ups, and to mediate communication throughout the design process.

The EuroCODE toolbox adds to Kyng's list the idea of deliberately working with contradiction as a vehicle for creative thinking, here realized not by merging but by contrasting the work-oriented and the technical checklists. Furthermore, it adds the idea of making theoretical knowledge available through checklists and prototypical examples. Still the overall argument for applying situational description in design concerns the social and cooperative nature of the design endeavour: representation in design is not a matter of holding on to something that is going - or not going - to be computed, as much as it is a matter of providing a common focus for discussing what, how and why something is going to be computed.

Though the framework was meant to serve the primary purpose of making shell and demonstrators available to people outside the project, it has found an equally challenging role in the project in finding a way of bridging between the empirical and theoretical concerns of CSCW and the more technical concerns.

Theoretical considerations about how to do this has been developed in [Bødker & Christiansen in press]. Here, we will move on to present the approach, before discussing it.

A CONCEPTUAL TOOLBOX IN CSCW DESIGN

The conceptual framework of EuroCODE is designed to deal with creative idea generation as well as systematic evaluation of ideas. In the conceptual toolbox pointers to theoretical insight from literature, and empirical knowledge from working with users on cooperative work and computer support for cooperation are put together and made available through checklists and examples. These may be revised by the designers. Furthermore, it is suggested how to use these design artefacts in scenario making. And by recommending scenario-making hand-in-hand with use of checklists we aim to support contradiction and dialogue.

Our inquiries into existing CSCW frameworks, e.g. [Schmidt & Bannon 92] have left us with the conclusion that they do not meet our requirements (see [Bødker, & Mogensen 93] because of their emphasis on defining concepts instead of demonstrating how work may become more cooperative by computer support. Thus, they are too little oriented towards change [Bødker & Christiansen in press] and do not sufficiently clarify how their concepts can be used in the process of designing CSCW applications, or where developers can find inspiration for good design ideas. Furthermore, they do not provide for the cultural, organizational and technological diversity, which is the norm in EuroCODE.

To meet the requirements listed in the introduction, the toolbox is consciously organized to let different perspectives talk to each other: Theoretical concerns are applied to focus the scenarios through **checklists**, which helps asking questions about a specific **work situation** and/or a specific **CSCW application**, thus enabling the designers to find out relevant constraints and key-concerns. We have initially developed a work-oriented checklist, taking into consideration aspects of overview, sharing, articulation work, etc. from literature and a technical checklist, taking up similar technical concerns, partly coming out of general experiences, and partly out of the kind of technical "material" that we work with in EuroCODE (Object oriented databases, hypermedia, etc.). Each item on the checklist points to a short (1/2 page) text elaborating on the theoretical and technical concerns. These checklists can be modified through use, and more checklists may be introduced (an overview of the interplay of these components is shown in Fig 1). The checklists will be presented in more detail later.

Provocation of thoughts and ideas, in scenario construction and evaluation is a matter of triggering ideas that are innovative on the one hand, but realistic and technically feasible on the other, recognizing the social, organizational and technical conditions which constrain a solution. A set of **prototypical examples** of CSCW technology serves to spark off good ideas for design and ground design considerations in practical experiences from literature or from the design domain. These prototypical examples have come out of applying one the one hand the work oriented checklist to constructs such as Sørgaard's shared material [Sørgaard 88], and Robinson's common artifacts [Robinson 93], and on the other hand the technical checklist to central applications of the EuroCODE shell, e.g. hypermedia.

The purpose of the scenario making activity is to support the creation of a universe for creative thinking, to force certain directions of thought, rules for actions to be taken within the universe, and a concrete example upon which to perform the dialogue among the involved actors. In the toolbox, different types of scenarios are suggested, and their components and internal structure is specified. This is in order to enable designers and users to efficiently capture and communicate their ideas.

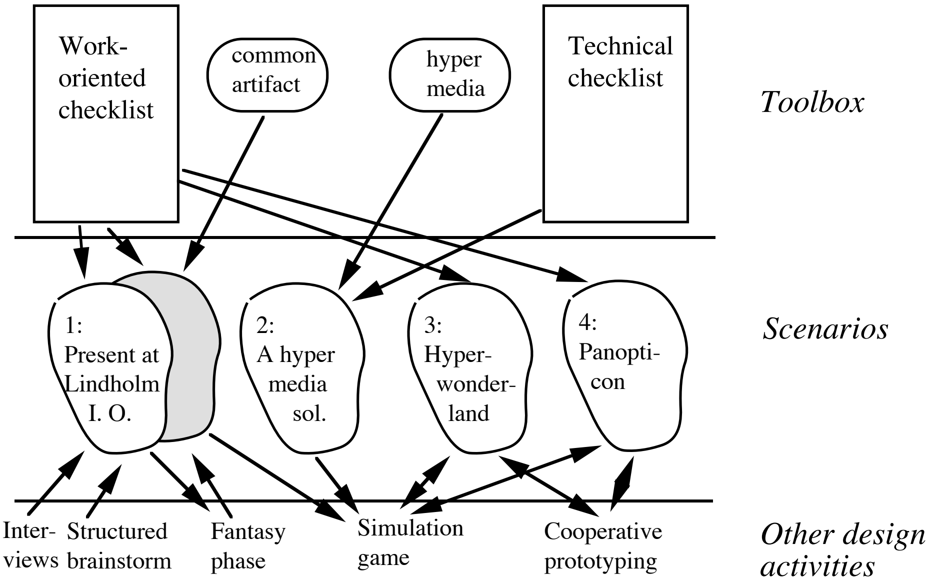

Fig. 1. The checklists and examples are used when creating scenarios, scenarios are the input as well as output of a number of other design activities. We have outlined here the example that we will discuss later.

APPLYING THE TOOLBOX - AN EXAMPLE

In the following we shall give an example of how we imagine the toolbox used in a design process at Great Belt. Though fictitious, the example replicates many of the experiences from using components of the toolbox in various settings in and outside the GB construction site, and it is based on empirical studies and cooperative design in EuroCOOP/EuroCODE. Furthermore, the actual kinds of scenarios suggested in the example are similar to those applied in EuroCODE at GB [Kyng in press, Grønbæk et al. 93].

The Great Belt bridge/tunnel project consists of a railway tunnel, a roadway bridge and a combined bridge. Great Belt Link Ltd. (GBL) was established to create the organization and to produce the material for the invitation of tenders. In a later phase GBL is supervising the construction activities. When the bridges and tunnel are completed in the late '90's, the organization will be responsible for operation and maintenance.

The construction is specified in thousands of pages. During construction these specifications evolve, and progress is monitored. In turn, this process involves thousands of pages of progress reports, change requests and non-conformance reports. The supervision involves three parameters: time, economy, and quality.

The design idea dealt with in our example is the following:

The supervisors at the construction site at Lindholm, the West Bridge, should be able to locate and annotate the documentation of the West Bridge, through a graphical representation of the construction site.

This design idea has come out of the EuroCoop/EuroCODE analysis and design work, which includes field studies of the work of supervisors, analysis of artifacts in use (filing facilities, construction drawings etc., etc.), future workshops with supervisors, etc. [Grønbæk et al. 93]. The following constitutes the fiction:

The specific problem was initially dealt with through a visit, where two designers spent two days at the construction site: after an initial tour of the Lindholm site, they conducted preliminary interviews with four supervisors. Initially, the work-oriented checklist had been used when creating the interview guide. On the second day of the visit, they each followed a supervisor around for the day. These initial investigations were used as outset for structured brain-storm in which the same four supervisors participated for one half day together with the designers[2]. The workshop was structured as the critique phase of a future workshop[3], where the participants took turns giving short statements of critique of their present work situation in the context of how they use documentation when on inspection.

Scenario 1: The Present at the Lindholm Inspection Office

This scenario addresses the problems regarding the present means of inspection, i.e. the filing and retrieval of construction documents, in particular regarding location and accessibility.

The setting is the Lindholm inspection office today.

Kurt is a supervisor at the Lindholm construction site. Together with 8-9 fellow supervisors he is supervising the prefab actions.

Kurt has access to a PC. The site office is connected to headquarters in Copenhagen. 'The Lindholm program' is an off-line print-out mapping the construction process. It is updated once or twice a week and is the easiest way to get an overview.

His office is located in one of the office-containers at the construction site. He and his colleagues are individually assigned to specific actions, but do co-operate. Kurt and John are assigned to caisson inspection. On Kurt's desk is a PC connected to the site office, which also serves as a meeting room for his group. Via the computer he can access his notes, some pictures etc., whereas he has to stay in contact with the secretary at the site office in order to get drawings, lab tests etc.

Action: Kurt is planning to leave his office to go out for an inspection.

While still in the container office, when he has decided which part of the production line to take a look at, he collects his notes and reaches out for the copy of the work procedure handbook for a check - just to realize that it is gone, probably because one of his colleagues has taken it along to the construction site.

Out on inspection, Kurt identifies some glide casting problems on the caisson that he is inspecting. He contacts the production engineer in charge and asks what steps have been taken so far. The production engineer claims that he has already written a memo and received acceptance of delay. Kurt regrets that he has not asked the secretary for a copy of the latest correspondence - now he has to go back to his office to get the documents.

Back in his office, he lays down his notes and phones the secretary, who asks for the identification number of the caisson under inspection. When telling her the number, he begins to doubt whether he has really inspected one of John's caissons as he had intended, or something else. He asks John and together they go to study the print out of ‘the Lindholm program’ hanging on the wall in the corridor.

Potentials, Problems and Bottlenecks:

Kurt faces problems with:

- the physical location of the documents needed;
- redundant archiving-the archives are both paper based and computer based, depending on location and version;
- different types of material, e.g. he relies on pictures and written text as well lab test results;
- re-finding material, as key-word entry quite often does not fit his current definition of a problem.

However, these problems may be overcome technologically by providing flexible read and write access to the master file of inspection.

Fig. 2, Scenario 1.

A scenario (Fig. 2) focusing on problems with the current work situation was made by the designers based on the input from the workshop. In the construction of the scenario, the work checklist has

[2] Similar to the processes described in [Bødker et al. 93a, and c, Bødker & Grønbæk 91 a and b].

[3] Future workshop activities described in [Kensing & Madsen, 91].

been used to raise critical questions to be covered by the scenario. The scenario is addressing the present, it is descriptive, similar to [Kyng in press]'s 'work situation descriptions and work situation overviews' that were actually used at GB.

The Process Continues...

An iteration of this scenario is the starting point of the fantasy phase of the future workshop, where the participants (the same four supervisors and two designers) work to imagine solutions to their problems, without too many concerns for the practical implementation of the solutions[4]. The common artifact example is used to help generate the fantasy. It is presented to the workshop by one of the designers. The work checklist is used by the designers to help consider the consequences of the choices made, resulting in a further scenario exploring primarily positive and negative trend situations in the future changed work, at an overall level. Furthermore, this scenario is explored in a simulation game between two designers and two supervisors[5], resulting in some modifications. A possible technical solution, starting out from the hyper media example (Fig. 3, Scenario 2) is used to explore the possibilities of in particular positive and negative trend situations (Fig. 4, Scenario 3 and 4), in a scenario similar to what [Kyng in press] calls 'exploration/requirement scenarios'.

Work-oriented and technical checklists are used to explore the consequences. Since they span the positive and negative sides of use of the same design suggestion, we have chosen to summarize the potentials, problems and bottlenecks for both scenarios at once. They formed the basis for, and were developed in iteration with, a cooperative prototyping activity[6], where an initial prototype was created by the designers, based on the revised technical scenario. Such scenarios have been used at the GB under the name of 'use scenarios' [Kyng in press]. The actual cooperative prototyping takes place in a setting at the Lindholm site, where two of the supervisors and two designers take part. The supervisors have been asked to bring relevant work material for an inspection task, the data of which exist also in the initial prototype. The scenarios set the stage and are discussed when problems mentioned in the scenarios occur, in the session. Finally, the potentials, problems and bottlenecks are discussed systematically when wrapping up the prototyping session.

As the solution in this case is accepted overall, the cooperative prototyping proceeds in three sequential half-day meetings with a more and more detailed and full-blown prototype. The process changes character from one where it is important to focus on the typical work around the prototype to one where it is important to explore and get hands-on experiences with a more and more complete prototype, covering more and more specific, yet peripheral work activities. The scenarios, in this process become increasingly rich, and the process changes to one where evaluation becomes more and more crucial. In the end, the refined technical scenario is used by the group of designers and users when handing over the prototype to those (including the two designers) who are to work on an implementation. (In Kyng's terms explanation scenarios). The two supervisors find it impor-

[4] How future workshops may be used this way is described in [Kensing & Madsen 91, Bødker & Grønbæk 91a and b].
[5] Similar to simulations described in [Bødker et al. 93a].
[6] [Bødker & Grønbæk, 91 b].

tant that all supervisors at GB are informed about the discussions of the initial design idea, and the scenarios form a good starting point for making such a presentation.

Scenario 2: A hypermedia solution

This scenario addresses the technical aspects of a hypermedia solution, applying portable PCs.

The setting is the Lindholm construction site sometime in the future.

Kurt has access to a portable PC. The portables are hooked up to the computer at the site office via a wireless modem connection, through which the supervisors run the hypermedia application.

Action: Kurt collects the drawing, he has received from the site office, takes his portable computer and goes out on inspection.

He stops at caisson no. 41 and checks the present state of affairs with the production plan and the drawings. Using the light pen on the bar code on the caisson he accesses the computer and gets a list of the latest correspondence on the screen. Clicking once more he sees at a glance that a delay has been accepted, and he starts a conversation with the production engineer about the progress achieved so far. Still on the spot, he enters his notes directly into the master file.

This way of working is made possible because the portables are hooked up to the site office. The interface is based on a graphical map of the construction site through which one can access a hypermedia application offering shared access to all kinds of material: drawings, notes, pictures, lab tests, non conformance reports, etc. By clicking on the graphical representation of the construction site, icons like caissons, cranes, and other relevant objects are accessible. These constitute the basic objects of the application. Links have been established to data about the accessible objects. By default, links go from objects to the master file, but it is also possible to make a link from the whole production line of caissons or to establish a link between a production line and a specific office at the site.

By clicking three to four times one can zoom in on details of the caisson under current inspection. By a single or double click one can get an overview of the relevant production plan or one can inspect who has recently checked the plans.

The process proceeds smoothly: a light pen is attached to the portable PC and, as all objects at the construction site have bar codes, it is easy to make the connection and get the relevant material right away. In addition, the location of other supervisors on inspection is monitored, provided they have a portable computer with them and are on-line.

Potentials, Problems and Bottlenecks - some examples

The physical distribution of data raises concerns about where the hypermedia objects are located, and to what extent they have to be transferred to the actual PC running the application.

The graphical representation of the work site will undergo changes as one caisson is finished, installed and replaced at the production line by a new one. The updating of the graphical representation including the initial establishing of links may be done by the supervisors on site, or as part of the work at the site office, where access to the master file may be more smooth and the time pressure less critical. This raises concerns for the updating of the shared database, for which objects are actually held in common by the users, and for the initial cost of setting up the user interface.

The graphical representation must represent all relevant objects in an overview and with a resolution that makes them accessible for mouse clicks.

Fig. 3. Scenario 2.

For the sake of the story here, we have assumed that a hypermedia solution is, at some level, a feasible solution to the design problem. Certainly, this does not have to be the case, though in the real case at the Great Belt there has been a lot of interest in hypermedia support for documentation [Grønbæk & Mogensen 94]. As a matter of fact, the whole point of user participation is to find out what needs to be done in the real work situation. The toolbox focuses on the kind of technical solutions that may be built using, and extending the EuroCODE shell. Cases where the EuroCODE platform needs to be abandoned all together is a different story.

Scenario 3: Hyper-wonderland

This scenario addresses the positive aspects of how a hypermedia solution will work.

The setting is the Lindholm construction site sometime in the future.

Kurt has access to a portable PC. The portables are hooked up to the computer at the site office via a wireless modem connection, through which the supervisors run the hypermedia application.

Action: During inspection of one of the caissons Kurt takes his portable PC, switches it on and places the cursor on the required information. He clicks the mouse button and gets the master file index together with an overview of links. He chooses the links of relevance for the caisson he is inspecting.

Kurt is pleased that he no longer needs to plan his inspections in advance. This is a great help because due to the 'event-driven' nature of inspection, constructors never know where and when an inspection is taking place. Moreover, it has become much easier to keep track of personal notes, reports etc. because they can be entered directly on the spot.

The access via the construction site interface does not force him to deal with complicated keywords either. Instead, he can access the relevant information right away, literally from where he is standing.

A positive side effect concerns his reachability. As long as he has logged in on the computer, he is within reach of the secretaries and can be contacted when guests arrive or when he is needed somewhere else on the site. Moreover, he can see at a glance where his colleagues are working and get in touch with them when he needs their help or advice.

All in all, Kurt feels that the new computer application has put him more in control of things.

Scenario 4: Panopticon

This scenario addresses the negative aspects of how a hypermedia solution will work.

The setting is the Lindholm construction site sometime in the future.

Kurt has access to a portable PC. The portables are hooked up to the computer at the site office via a wireless modem connection, through which the supervisors run the hypermedia application.

Action: During inspecting one of the caissons Kurt starts talking to one of the builders about some reinforcement problem. They argue about the recent lab tests, and he takes out his portable PC in order to provide some data which justify his arguments. It takes quite a while before he finds a spot where he can place the PC: either there is too much light, or there is no level surface at a suitable height. Finally ,he puts the laptop on a big box and switches it on. He positions the cursor on the caisson he is currently inspecting and clicks the mouse to get into the master file. The table of contents pops up and from the overview of links he chooses those of relevance - but no lab test appears on the screen. Obviously, the file has not been updated as planned.

Kurt is rather upset. This loss of prestige in front of a contractor engineer would not have happened if he had planned his inspection as he had in the old days.

Sometimes, he feels like a hunted fox especially in situations where he is drifting around thinking about what kind of action to take in a particular case. If he has forgotten to log out, he suddenly has a secretary on the phone: "I see you are right at caisson 39, so could you not just drop by and take a message?"

All in all Kurt feels that the new computer application has put him under control.

Potentials, Problems and Bottlenecks:

Indeed, more information becomes readily available with the discussed solution. Yet the portables need attention to be usable: they need to be reliable and the database updated. Furthermore, the inspectors need to consider how they may place the computers in the field. Training and procedures to support smooth ways of including the computer applications in the inspection work should be investigated.

It should be discussed how much preparation of an inspection may be substituted by in situ retrieval of documents.

Reachability and monitoring are flip-sides of the coin and need to be debated as well.

Fig. 4. Scenarios 3 and 4.

THE TOOLBOX - A SYSTEMATIC PRESENTATION

The checklists that were applied in the example are presented in fig. 5 and 7, and examples of the detailed description of items in fig. 6 and 8. Descriptions of items start with an introduction, then tell its user what to consider and finally list a couple of specific issues to be checked.

Items of the work-oriented checklist are put together as pointers to sociological CSCW literature [Schmidt & Bannon 92, Hughes & King, 92, Heath & Luff 92, Hutchins 90, Robinson 93, Star & Griesemer 89]. Each item of the work-oriented checklist captures a specific part of the work situation and thus opens up a unique perspective on the workplace. These perspectives may be redundant since some items on the list overlap, e.g. details on the physical workplace may include similar information as details on the spatial distribution of actors and tools.

The work-oriented checklist consists of fourteen items for capturing crucial aspects of work situations. These items address:

1. the physical setting of the workplace
2. the organizational environment
3. the actors in the work situation
4. the working activities and their relations
5. the materials used and the outcomes produced
6. the communication facilities and tools
7. the standards for procedures and products
8. the security measures and access rights within the organization
9. the sequentiality of procedures and (sub-) products
10. the actors' awareness of each others' activities
11. the actors' overview of current projects and tasks
12. the spatial distribution of actors, materials and tools
13. the degree and kinds of mobility required from the actors
14. the constraints of current working conditions and possibilities for changing them.

Fig 5. The work-oriented checklist.

Materials and outcomes:

In the course of activities, different kinds of materials and information may be required as input. These are processed by the actors and may lead to various intermediate results and/or final products.

Consider which different kinds of material, intermediate results and final outcomes are crucial for the work situations you are investigating:

- What kind of materials (documents, pictures, drawings etc.) are involved in the production of which kind of products?
- Which sub-products or intermediate results are produced?
- How are the sub-products or intermediate results used in further production, i.e. what are the crucial "hand overs" and transitions, and when and how do they take place?
- What are the metaphors currently used to characterize materials and results?

Fig 6. Detailing item 5 on the work-oriented checklist.

Redundancy of that kind is incorporated into the checklist for three reasons:

- It helps increase the completeness of work aspects relevant for designing a CSCW solution. For example, if details about an actor's physical location are missing from the description of the physical setting in the first item there is a good chance that they are mentioned in item 12 or even 13.
- It helps achieve new insights into particular characteristics of cooperative process since it addresses the same aspects from different points of view. For example, the creation and continuous manipulation of a common artifact may be seen from the perspective of activities and their relations (item 4) or from the perspective of materials, successive sub-products and final outcomes (item 5). Both may have different implications for providing technical support.
- It helps detect contradictions, inconsistencies or hidden aspects. For example, actual procedures and actors mentioned in item 3 and 4 may deviate from organizational standards addressed in item 7. Such contradictions indicate where things are still unsettled, unclear or controversial thus may point to potentials for reorganization and optimisation.

A similar kind of redundancy is found in the technical checklist, where redundancy may trigger conflicting design ideas thus hinting at the creative space for technical solutions. The technical checklist addresses important features of technical solutions - in particular CSCW solutions - that should be considered in designing systems and applications.

The technical checklist consists of twelve items for capturing crucial aspects of technical solutions. These items address:

1. the metaphors to be used for constructing or understanding the technical solution
2. the major, object-oriented concepts
3. the physical location of hard- and software
4. the complexity of the technical solution
5. the sequentiality of activities enforced
6. the extent and forms of functional integration
7. the security measures and access rights provided
8. the CSCW potentials which help users to overcome distances in time and space
9. the potentials for awareness support
10. the extent and kinds of tailor ability
11. the cost benefit distribution with respect to different cooperating users
12. the least effort strategies with respect to implementation, introduction and maintenance.

Fig. 7. The technical checklist.

Metaphors as springboards:

Metaphors designate a comparison between things essentially or generically different but strikingly alike in some pertinent aspects. They can be used to characterize technical facilities, to support their understanding and to generate surprising technical solutions. Therefore, they may serve as springboards for new design ideas [Madsen, 87]

Consider which metaphors are appropriate to characterize and illustrate a technical solution or components of a technical system:

- How can the technical solution (or parts of it) be "seen as something else" or can be "described in terms of something else"?
- What are the correspondences between the metaphor and the technical solution; where are limits and differences?
- In particular, which metaphors can be used for illustrating the user interface, data structures, or particular functionalities?

Fig 8. Detailing item 1 on the technical checklist.

Items on the list may provoke multiple or even competing design ideas. For example, when designers discussed the first item on the checklist for the hypermedia component of the EuroCODE environment, they produced a variety of metaphors, such as "a book", "a trail leading through information", "a kaleidoscope", and "glued material" [Bødker et al. 93b]. Obviously, each of these metaphors may trigger different design ideas, e.g. for the user interface. Thus, the technical checklist applied by several members of a design team helps to capture multiple ideas and can serve as input for presentation, discussion and decision making.

As prototypical examples we have crystallized a series of work-oriented examples, as well as technical ones to think from, and kick off new ideas. These examples are structured through the use of the checklists. It would take things too far to go through all of the examples systematically here. Nevertheless, we have briefly summarized one of the work-oriented examples (Fig. 9), as well as a technical one (Fig. 10).

Cooperation through Communication

According to [Robinson 93] a common artifact is an elaboration of the dimensions of communication that take place through, and are supported by a computer application / an artifact. Any specific artifact with the requisite dimensionality is considered a common artifact. A common artifact is an effective tool for getting a job done; it helps people see at a glance what others are doing; it enables actions and changes made by others to be understandable, and appropriate changes to be made; it provides a focus for discussion of difficulties and negotiation of compromises; it offers an overview over the work process that would not otherwise be available.

Four dimensions of common artifact are described:

• predictability. This covers issues like dependability, functionality (incl. consistency of, compatibility of),and appropriate interface.

• peripheral awareness conceived as local and immediate ('at a glance')

• double level language: includes conventionalised implicit communication through the artifact ('shared material') and the role of artifact as 'indexical focus' for dialogue.

• overview: this is the complement to peripheral awareness. A common artifact can make situations outside the here and now, available within the here and now.

Robinson gives examples such as a hotel key rack and a map. [Hutchins 90] has described how a map can function as a common artifact for a group navigating a large ship. In both cases the physical setting in terms of where the artifact is placed is important for who has access to its information when.

The common artifact is a tool and may at the same time serve co-ordinating purposes by making people's activity visible to others: by doing so, the common artifact all in all serves as a mediator of the cooperation that takes place. The mediating functions may be embedded or open to communication depending on how stable the setting, the problem space and the actors' knowledge and skills are.

A common artifact does not necessarily enforce sequentiality, but it may offer it: e.g. you have to assume that all keys not in use have been replaced, in order to be correct in interpreting an empty hook to mean that the guest is in.

The question of robustness to unanticipated use is important here, due to the high probability that users will depart from any sequence of operations that can be anticipated. In the case of a map, the use is quite open to differences in sequencing.

The concept of a common artifact springs from experiences with co-located work settings, yet the access to the artifact may be distributed in time and space. E.g. the users do not have to check in on the hotel key-rack simultaneously. Also those users who only need a glance at the key-rack may do so from a distance, or even via remote connections such as video. In keeping with the above discussion, a pure video connection would probably not be sufficient as the only access to an artifact.

Fig. 9. A work-oriented example.

Hypermedia

A book is a good metaphor for hypermedia. The materials of a hypertext are similar to those of a book: texts, pictures and graphics. Contrary to the book, a hypertext may contain e.g. video and sound.

A book facilitates unstructured browsing through the materials, as well as it provides structured means for retrieval of materials. In the book, as opposed to hypertext, these structures cannot be changed once the book has been put together.

A kaleidoscope, another useful metaphor, provides new patterns in the material by seeing it in no predetermined, yet systematic order. The kaleidoscope allows the user to find new trails through the material. With the hypertext, as opposed to the kaleidoscope, the trails need to be defined before use.

None of these metaphors tell much about how the structure of a given hypertext comes into being.

Hypermedia technology (hypertext) has shown promise in supporting groups working on shared materials. Hypermedia technology helps maintaining networks, hypertexts of associations, links or link components, between chunks of materials, nodes or components. Anchors inside materials may be supported, for instance text components may provide anchoring of single characters, words and paragraphs as endpoints for links. Anchors appear in the interface as link markers.

Hypermedia clients and database servers may be distributed over several machines in a local area (and/or wide area) network. The database notifications carry information about which machine the connected users are using when modifying hypertexts and components. Hypermedia facilities may also become available for mobile computers.

Hypertexts as well as any individual parts of them can be held in common by several users, though components may be temporarily locked during a linking procedure. Designers need to decide the granularity for locking.

Hypermedia applications usually only support asynchronous cooperation.

The event notification mechanism supports users in maintaining awareness of who is doing what on the shared materials. The hypermedia client provides support for propagating notifications as a combination of messages in a console, bell sounds, small icons appearing in editor or browser windows and as immediate updates.

Fig. 10. A technical example.

Summarizing the example gives an idea about how the various components of the toolbox may be systematically applied. Fig. 11 looks at the various cooperative design activities in the example, emphasizing which components were used in the activity, what were the outcome, and who took part. The figure illustrates that scenarios take on the typical double role of design artifacts, of being produced in some activities, and instruments of others. Fig. 12 characterizes the use of the toolbox components as design artifacts, in exactly these roles.

Activity	Artifacts used	Outcome	Participants
Interviews and field trip	Work-oriented checklist	Scenario 1	designers as producers
Future workshop, critique	Scenario 1, work-oriented checklist	Scenario 1	users and designers
Future workshop, fantasy	Scenario 1, work-oriented checklist, common artifact example	Scenario 3 and 4	users and designers
Technical scenario production	Hypermedia example, technical checklist	Scenario 2	designers as producers
Simulation game	Scenario 3 and 4, scenario 2	Scenario 3 and 4	users and designers
Cooperative prototyping	Scenario 3 and 4, scenario 2 Prototype	Scenatio 3 and 4, prototype	users and designers

Fig. 11. The example summarized regarding the roles of checklists and scenarios.

Design artifact	Description	Produced in or as a result of e.g.	Produced by (typically)	Used in or provides basis for	Used by (primarily)
w-o checklist	raises questions about typically critical aspects regarding work, in particular cooperation and computer support for this	framework construction based on theory revised based on experience	metaactivity + designers	analysis of existing work situations, scenario production, exploring scenarios	designers
technical checklist	raises questions about typically critical aspects regarding technical implementation, in particular regarding sharing and distribution of technical solutions	do + EuroCODE technical possibilities	metaactivity + designers	investigation of existing computer application or technical design suggestion exploring scenarios	designers
w-o prototypical example	presents examples of work organizational constructs to enhance computer supported cooperative work - potentials and problems	do	metaactivity + designers	idea generation scenario production	designers
technical prototypical example	present sexamples of technical constructs to enhance computer supported cooperative work - potentials and problems	do	metaactivity + designers	idea generation scenario production	designers

Fig. 12. Applying [Kyng in press]'s format for characterizing design artifacts, on the toolbox components. We see these mainly as tools for designers.

Scenario-making

In general, scenario construction is an important method for assessing mid- and long-range developments in technology, economy and society. Scenarios allow for the description of alternatives and may address current situations as well as hypothetical ones in the future. What is characteristic for our way of thinking about scenario construction is that they are constructed in tight loop with their use in groups of designers and users - perhaps users do not always take part in the initial construction of scenarios, but they do in the iteration that takes place in joint design activities such as prototyping sessions or future workshops. Scenarios, as any other design representation, serve the double purpose of engendering the decisions made in the design situation, and of being a medium of communication between the persons involved in the design activity, most noticeable in our context are the cooperating designers and users, but also people from outside the group, e.g. future users or potential purchasers of the computer application and other groups of designers.

The term 'scenario' is used with various meanings in the literature. These conceptions, however, share several features that more or less constitute the term: A scenario is *hypothetical*, i.e. it describes some possible or potential alternative in the present or future. It is *selective*, i.e. it represents one possible state of complex, interdependent, dynamic and opaque affairs; *bound*, i.e. it consists of a limited number of states, events, actions and consequences, or subsets of these categories; *connected*, i.e. its elements are conditionally, temporally or causally related; and *assessable*, i.e. it can be judged with respect to its probability and/or plausibility.

Because of these features, scenarios leave enough freedom for a 'disciplined intuition', but constrain the construction process in a reasonable way: The demand to be hypothetical but possible distinguishes scenarios from mere science fiction stories. Selectivity and boundness imply that a single scenario should not try to capture an extensive and complex domain. If such a domain needs to be handled it should be broken down and addressed by a set of scenarios. This claim has consequences for the size of scenarios, i.e., they should be concise stories instead of long-winded novels. Connectivity suggests that a scenario should form a coherent entity in terms of stories which represent conditional, temporal or causal relationships between crucial states of affairs. Accessibility, at last, ensures that a scenario can be evaluated to find out how it is constrained. Scenarios should be 'stories' located in time and space, 'traces' featuring details, not 'novels'. The scenario should capture the main persons and their activities. This information should allow for conclusions which may serve as answers to the issue of the scenario. Hence, a scenario should consist of at least four parts: an issue, a setting, a story and a conclusion.

[Campbell 92] categorizes various types of scenarios dealt with in the HCI literature, based on the assumption that a scenario refers to "representative instances of interaction between user and system". He mentions:

- Scenarios for *illustration* aim to clarify what it is like to use a system.
- Scenarios for *evaluation* take the form of evaluation tasks by specifying step-by-step procedures to be carried out by users in the evaluation of a system.
- Scenarios for *design or redesign* should represent examples of user-system interaction and may contain correct as well as faulty user activities.
- Scenarios for *testing theories* investigate the accuracy or predictive power of a theoretical approach in HCI.

Scenarios may also be the basis for illustrating ways of introducing a computer application into an organization, strategies for marketing and education.

Many characteristics of scenarios depend on their use. For example, [Carroll et al. 91, Carroll & Rosson 92] introduce the notion of critical and typical situations: Scenarios should be designed based on knowledge about typical ways of doing things, but addressing specific, critical instances of the typical. This distinction is analogous to the one by [Ducot & Lubben 80] who use the terms 'trend' versus 'peripheral' and moreover differentiate between descriptive and normative scenarios. In addition, it is useful to distinguish between scenarios which describe present or future situation and those which focus on positive or negative aspects. When these characteristics are combined they constitute different types of scenarios which are applied differently in the design process (Fig. 13).

Peripheral or critical scenarios may include situations that are contradictory to the mainstream. And actually applying the checklists may result in contradictory scenarios or perspectives. This is not a problem, as we see it: Contradictions are thought provoking in the interaction among designers and users and imply that there often are not one best solution, in contrast to the assumptions of many traditional systems development methods. The creative space is exactly where contradictions are confronted.

Scenario types	Description	Produced in or as a result of e.g.	Produced by (typically)	Used in	Used by (primarily)
Present work descriptive scenario (scenario 1)	summarizes the way current work in the organization is interpreted by the design team, centered around relevant, *existing* situations, within the users' workplace - duely reminded by the theoretical questions from the work-oriented checklist	initial study, work-oriented checklist	designers	e.g. Future workshops, development of use scenarios and mock-up/prototypes	designers and end-users
Future work scenarios overall	indicate how computer support and (or) changes in work organization in the future may improve upon work situations, and set the stage for simulation at workshops	embodying ideas and workshop preparations, technical and work-oriented checklist w-o prototypical examples	designers and end-users	e.g. simulation games	end-users
Future work scenarios detailed positive or negative (scenario 3 and 4)	detailed scenarios supplying the use-details needed for discussing whether or not suggested technical capabilities will function in use explain/hypothesize about new possibilities for support with the current prototypes	embodying ideas and workshop preparations, technical and work-oriented checklist prototyping	designers and end-users	e.g. prototyping, mock-ups	end-users
Future technical scenarios (scenario 2)	detailed scenarios supplying the use-details needed for discussing whether or not current technical capabilities meet the requirements of the scenario	explorations of capabilities of existing software designs technical checklist technical prototypical example	designers	discussions of capabilities of existing software designs	designers and end-users

Fig. 13. Scenarios as design artifacts.

In [Bødker & Christiansen in press], we discuss, as mentioned in the introduction, how our CSCW toolbox may be thought of from a theoretical point of view. We discuss how we can view CSCW design artifacts as support for bridging people and matters that are physically, timely and conceptually apart; they must be open and negotiable and facilitate dialogue around contradictions.

Certainly the notion of, and reasons behind, the toolbox have to be rethought theoretically and practically based on the experience we collect from practical use and revision of the toolbox. In the following we indicate the lines along which we expect to gain input for an iteration.

APPLYING THE TOOLBOX

The toolbox may be applied in many different ways and for a variety of purposes. A few examples are:

- to use the work-oriented checklist for constructing questionnaires, guiding observations, and structuring collected data;
- to use the technical checklist for presenting design ideas in an informal but well structured way to users, other designers, potential customers etc.;
- to use the prototypical examples as a starting point for considering solutions and as an input for discussing their adaption and enhancement;

- to use the different kinds of scenarios for focusing on specific aspects of a present work situation or for describing the potential impact of a CSCW solution on a working environment.

So far, we have deliberately avoided to specify procedures of how to use the toolbox, and the framework does not propose or constrain how to combine its components with other techniques. Our example above gives some indications, however, of the type of process we are after:

1. A series of scenarios constitute the back-bone of the design project.
2. These scenarios help span out a theory-oriented exploration of design situations, while keeping the grounding in the specific empirical setting.
3. Numerous activities take place around the design-activity, ranging from initial interviewing and observation to programming and testing.
4. Some of these activities, as well as the actual shaping of the scenarios, are cooperative activities among users and designers, others are primarily done by professional designers.
5. Since the scenarios as such do not embody the technology, thus not being available for hands-on experiences by users, scenario making will primarily form the basis for other activities in which they are embodied and explored. They may be setting the stage for and pointing at problems and solutions to be dealt with in cooperative prototyping [Bødker & Grønbæk 91a and b], when using mock-ups [Ehn & Kyng 91], simulations, or in more systematic explorations of a running computer application for evaluation or education purposes.
6. Furthermore, the scenarios can be designed and explored in other types of design-by-doing situations such as future workshops [Kensing & Madsen 91], organizational games [Ehn & Sjögren 91], dilemma games [Mogensen 94].

As mentioned, our experiences with scenarios as useful artifacts in cooperative design situations date back to the EuroCoop project [Kyng in press] and even earlier [Bødker 91, Jungermann & Thüring 87]. However, practical experiences with the use of the toolbox are still limited. A first version of the toolbox has been introduced to the EuroCODE community in a workshop and used by three groups of designers in this context. Based on feedback from this, the checklists were revised, and the internal structure for scenarios was described in more detail. The toolbox will be continuously improved and revised in the further course of the EuroCODE project, in iteration with the development of the EuroCODE shell and demonstrators, thus hopefully supporting the creative elements in EuroCODE itself.

ACKNOWLEDGEMENTS

The work has been funded in part by Esprit project 6155 - EuroCODE. The EuroCODE T1.1 group (Kaj Grønbæk, Morten Kyng, Kim Halskov Madsen, Preben Mogensen, Peter Axel Nielsen and Mike Robinson of Aarhus University; Heike Kühn and Simon Robinson of empirica; Elke Hinrichs and Thomas Kreifelds of GMD; Pål Sørgaard, NR; Pippa Hennesy, Nexor; Lone Faber, Daniele Pagini and Wendy Mackay of RANK Xerox EuroPARC) helped develop the toolbox. Our presentation owes much to Morten Kyng. Olav Bertelsen, Tom Moran and several anonymous reviewers provided useful comments on earlier drafts. We thank Susanne Brøndberg for improving our English, and Olivier Danvy for the French translation of the abstract.

REFERENCES

Bowers, J . & Pycock, J. (1994). Talking through design: Requirements and resistance in cooperative prototyping, Adelson, B., Dumais, S. & Olson J. *CHI '94 confernece proceedings*, ACM, New York, NY.

Bødker, S. & Christiansen, E. (in press). Scenarios as springboards in design. In Bowker, G., Gasser, L., Star, S.L. & Turner, W. (eds.), *Social science research, technical systems and cooperative work*.

Bødker, S. & Grønbæk, K. (1991a). Design in Action: From Prototyping by Demonstration to Cooperative Prototyping. In Greenbaum, J. & Kyng, M. (eds.), *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, N.J.: Lawrence Erlbaum Associates, pp. 197-218.

Bødker, S. & Grønbæk, K. (1991b). Cooperative Prototyping: Users and Designers in Mutual Activity. *International Journal of Man-Machine Studies*, Special Issue on CSCW, 34, pp. 453-478.

Bødker, S. & Mogensen, P. (1993). One woman's job is another man's articulation work - an essay about the design of computer support for cooperative work. In Robinson, M. & Schmidt, K. (eds.), *Developing CSCW Systems: Design Concepts. Report of the CoTECH WG4*, pp. 149-166.

Bødker, S. (1991). *Through the Interface – a Human Activity Approach to User Interface Design.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Bødker, S., Christiansen, E., Ehn, P., Markussen, R., Mogensen, P. & Trigg, R. (1993a). *The AT project. Practical research in cooperative design*. DAIMI PB-454, Department of Computer Science, Aarhus University.

Bødker, S. et al. (1993b). *Deliverable D 1.1: The EuroCODE Conceptual Framework*: Preliminary, empirica, Bonn.

Bødker, S., Grønbæk, K. & Kyng, M. (1993c). Cooperative Design: Techniques and Experiences from the Scandinavian Scene. In Namioka, A. & Schuler , D. (eds.), *Participatory Design: Perspectives of Systems Design. Principles and Practices*. Hillsdale, N.J.: Lawrence Erlbaum Associates, pp.157-76.

Bødker, S., Ehn, P., Kammersgaard, J., Kyng, M. & Sundblad, Y. (1987). A Utopian experience. In Bjerknes, G., Ehn, P. & Kyng, M. (eds.), *Computers and democracy: A Scandinavian challenge*. Aldershot, UK: Avebury, pp. 251–278.

Campbell, R.L. (1992). Will the real scenario please stand up? *SIGCHI Bulletin*, 24 (2), pp. 6-8.

Carroll, J. M. & Rosson, M.B. (1992). Getting around the task-artifact cycle: how to make claims and design by scenario, *CACM*, 10(2),pp. 181-210.

Carroll, J. M., Kellogg, W.A. & Rosson, M. B. (1991). The task-artifact cycle. In Carroll, J. M. (ed.) *Designing Interaction: Psychology at the Human-Computer Interface*, New York: Cambridge University Press, pp. 74-102.

Ducot, C. & Lubben, G.J. (1980). A typology for scenarios, *Futures*, 12, pp. 51-57.

Ehn, P. & Kyng, M. (1991). Cardboard Computers: Mocking-it-up or Hands-on the Future. In Greenbaum, J. & Kyng, M. (eds.), *Design at Work: Cooperative Design of Computer Systems*, Hillsdale, N.J.: Lawrence Erlbaum Associates, pp.169-195.

Ehn, P. & Sjögren, D. (1991). From System Description to Scrips for Action. In Greenbaum, J. & Kyng, M. (eds.), *Design at Work:Cooperative Design of Computer Systems*. Hillsdale, N.J.: Lawrence Erlbaum Associates, pp. 241-268.

Ehn, P. (1988). *Work-oriented design of computer artifacts*. Falköping: Arbetslivscentrum/Almqvist & Wiksell International, Hillsdale, NJ: Lawrence Erlbaum Associates.

Engeström, Y. (1987). *Learning by expanding*. Helsinki: Orienta-Konsultit.

Grønbæk, K., Kyng, M. & Mogensen, P. (1993). CSCW Challenges: Cooperative Design in Engineering Projects, *CACM* vol. 36 no. 4, pp. 67-77.

Grønbæk, K. & Mogensen, P. (1994). Specific cooperative analysis and design in general hypermedia development. In Trigg, R., Anderson, S. I. & Dykstra-Erikson, E. (eds.), *PDC '94: Proceedings of the Participatory Design Conference*, Palo Alto, CA:CPSR, pp.159-172.

Heath, C. & Luff, P. (1992). Collaboration and control. Crisis management and multimedia technology in London Underground line control rooms, *CSCW Journal* Vol. 1 no. 1-2, pp. 95-118.

Holmes, P. (1994). Hypermedia, Desktop conferencing and PACS. *Proceedings of EuroPACS - The 12th International EuroPACS meeting*; Geneva Switzerland, September 22-24, 1994.

Hughes, J. & King, V. (1992) *Paperwork*. COMIC Working Paper COMIC-LANCS-4-1, Lancaster University.

Hutchins, E. (1990). The Technology of Team Navigation. In J. Galegher, R. E. Kraut & C. Egido (eds.), *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work,* Hillsdale, N.J.: Lawrence Erlbaum Associates, pp. 191-220.

Jungermann, H. & Thüring, M. (1987). The use of mental model for generating scenarios. In G. Wright & P. Ayton (eds.), *Judgmental forecasting*. New York: John Wiley and Sons, pp. 245-266.

Kensing, F. & Madsen, K. H. (1991). Generating Visions: FutureWorkshops and Metaphorical Design. In J. Greenbaum & M. Kyng (eds.), *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, N.J.: Lawrence Erlbaum Associates, pp.155-168.

Kyng, M. (in press). Creating Contexts for Design. In Carrol, J. (ed.), *Scenario-Based Design For Human-Computer Interaction*, John Wiley & Sons, 1995

Madsen, K. H. (1987). Breakthrough by Breakdown. In Klein, H. K. & Kumar, K. (eds.), *Proceedings of the IFIP WG8.2 Working Conferenceon Information Systems Development for Human Progress in Organization*, Atlanta, 29–31 May 1987, Amsterdam: North-Holland.

Mogensen, P. (1994). *Cooperative analysis*, Ph. D. thesis, Aarhus University.

Robinson, M. (1993). Design for Unanticipated Use. In G. deMichaelis & C. Simone (eds.), *Proceedings of the Third European Conference on Computer-Supported Cooperative Work* (ECSCW '93). Dordrecht, Boston, London: Kluwer, pp. 187-202.

Schmidt, K. & Bannon, L. (1992). Taking CSCW Seriously: Supporting Articulation Work. *Computer Supported Cooperative Work* (CSCW), 1 (1), pp. 7-40.

Star, S.L. & Griesemer, J.R. (1989). Institutional Ecology,'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley’s Museum of Vertebrate Zoology, 1907-39. *Social Studies of Science* 19, pp. 387-420.

Sørgaard, P (1988). Object Oriented Programming and Computerised Shared Material. In Gjessing, S. & Nygaard, K. (eds.), *Second European Conference on Object Oriented Programming (ECOOP '88)*, Springer Verlag, Heidelberg, pp. 319-334.