Groupware Task Analysis and Distributed Cognition: Task Modeling In a Case of Multiple Users and Multiple Organizations

Mari Carmen Puerta Melguizo*, Cristina Chisalita[§] &

Gerrit C. van der Veer[§]

Institute of Information and Computing Sciences. University of Utrecht (The Netherlands).

[§]Section Human-Computer Interaction, Multimedia and Culture. Department of Information Management and Software Engineering (IMSE).Vrije Universiteit, Amsterdam (The Netherlands).

Introduction

"Complex interactive systems" are information systems used by different types of users for a variety of tasks and in different situations. Work activities in these cases usually include communication and co-ordination between people and actions of several persons on shared objects and in shared workplaces.

In many cases the (re)design of these type of systems is triggered by an existing task situation. Either the current way of performing tasks is not considered optimal, or the availability of new technology is expected to allow improvement over current methods. A systematic analysis of the current situation reveals problems, conflicts and inconsistencies that occur while performing the tasks. Task analysis may help to formulate design requirements, and at the same time it may later on allow the evaluation of the design. In general, analyzing the current situation means to analyze the "context of use" or, in other words, the world in which the system functions. According to the ISO 9241-11 standards [7] this means to analyze the users, the tasks, the equipment (hardware, software, and materials) and the environment. In the case of complex systems, aspects such as social, historical and cultural context of the human activity are of especial relevance and need to be analyzed as well.

R. Navarro-Prieto and J.L. Vidal (eds.), HCI Related Papers of Interacción 2004, 119 -135.

We are interested in knowing how such complex situations can be best described and analyzed. A possible point of departure is represented by studies focused on the description of simpler situations (e.g. the cooperative work that takes place in settings such as operators' rooms). Such studies suggest that exploring and comparing different frameworks brings important conclusions about the complementary information each framework can bring [2]. Therefore, we decided to use two different frameworks: GroupWare Task Analysis (GTA) and Distributed Cognition (DC) to analyze a case study of a complex situation. In the next sections we will explain briefly both frameworks and show the descriptions we obtained applying both approaches in the analysis of a highly complex system.

GTA: GroupWare Task Analysis

GTA is a broad task-analysis conceptual framework that is based on the integration of several approaches including individual oriented method from Human-Computer Interaction and group oriented methods from Computer Supported Cooperative Work [12, 13, 14, 15]. GTA describes the task world focusing on three different viewpoints:

1. Agents and roles. Specifying the active entities in the task world including users and stakeholders, systems and organizations. The roles and the organization of work (i.e. structure of agents and roles) need to be specified as well. Agents can be characterized on relevant characteristics such us a) psychological characteristics like cognitive styles or spatial ability [12], and b) task related characteristics like expertise or knowledge of information technology.
2. Work. Specifying the decomposition of tasks, the goals, the events that trigger the tasks, and the different strategies used to perform them.
3. Situation. Specifying the objects used in the task world as well as their structure, the history of past relevant events, and the whole social and physical work environment. Objects may be physical or conceptual (non-material: like messages, gestures, passwords, stories, or signatures).

Modeling the task knowledge is not an easy activity. There are several problems that may arise in such situation (e.g. the amount of data can be overwhelming and difficult to organize). In order to overcome these problems the GTA framework describes a task world ontology that specifies the relationships between the concepts on which the task world is

modelled (see figure 1). Based on this ontology a supporting tool to model task knowledge was also developed: EUTERPE [1, 15].

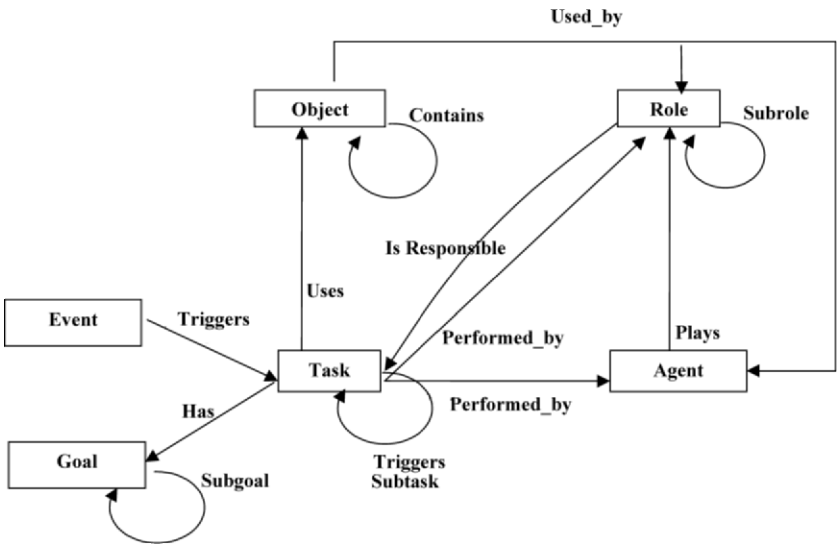

Fig. 1. The task world ontology

DISTRIBUTED COGNITION

Distributed Cognition (DC) is a framework that describes and explains group cognition with the goal of understanding how collaborative work is coordinated. DC takes as unit of analysis the *socio-technical system* (Hutchins [5] calls it the *cognitive system*) and investigates the shared construction of knowledge. The basic assumption is that the socio-technical system consists of multiple individuals as well as the artifacts they work with, and the cognition is distributed between them. According to this approach, cognition is the process of coordinating the distributed internal representation (i.e. in the minds of people collaborating together) and the external representations (i.e. information artifacts). DC studies the way in which the information and the knowledge is transmitted, transferred and transformed through the different representations during system's activities. DC has been applied as a framework to study the cooperative activity in different settings [3, 4, 5, 6, 9, 10].

To analyze the distribution and communication of knowledge, the different interactions that take place within the socio-technical systems are described [8] (see figure 2):

1. Social Distribution: interaction among people.
2. Technological Distribution: interaction among people and artifacts.
3. Interaction among people, artifacts and work environment.

The elements involved within the socio-technical system (individuals and artifacts) hold two types of knowledge [11].

- *Distributed or different knowledge* about the task/activity. Each individual has specific knowledge which represents a part of the knowledge needed in order to perform the task/activity.
- *Shared knowledge* about the task/activity. Each individual shares a part of the knowledge necessary to perform the task. This shared or common knowledge allows the performance of the task to be monitored by different individuals. For example, in the case of individual failure, another member of the system can perform the task and in this way, the system does not need to interrupt its activity.

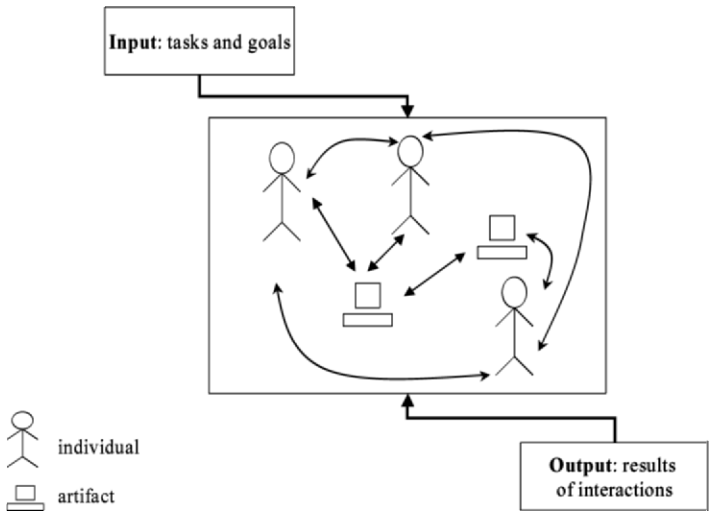

Fig. 2. The socio-technical system

The Task Analysis of a Highly Complex System

The case study that we describe in this paper shows our experience collaborating with a company that produces high-tech machines. The machines are sold to different industries in several countries of Europe, America and Asia. The machines produce very fragile and expensive products that are complex and difficult to use. As a consequence, many errors and problems occur while using the machines. In order to help its customers, the design company offers assistance services provided by different types of engineers. However, users complain frequently, and especially, about the user interface that is difficult to understand and use. Following these complains the company decided to take action and to find new solutions.

We were invited to collaborate with the User Interface Design team of the company. The goals of the design project were: a) to integrate different versions of the same interface (differences were related with the "age" of the systems and the fact that newer versions contain more functions than the first ones), b) to support all types of users who are using the interface and, c) to design and integrate in the interface a consistent "diagnostic toolkit". The toolkit should be able to identify the cause of problems while using the machines and provide a solution. The last aspect is the main focus of this paper.

The first step in the project was to analyze the context of use. We used several techniques to do that: interviews, "focus-group" sessions (in which people from different departments of the manufactory were invited to identify and define the types of users that use the system), documents of the company (including a previous document on task analysis), etc. The task analysis proved to be a very complex process by itself because the tasks and the technology to consider are highly specialized. Because we cannot reveal the identification of the company we will not show a detailed task analysis.

Using GTA: Multiple Users

Following GTA we started to analyze and model by defining the *agents*, the *work* and the *situation.*

Actors, Roles and Tasks

Many different types of users work with the system including end-users (Operators that are the employees of the customer company) and several types of expert's engineers. In total more than 40 different actors were defined (e.g. field application engineers, system packers, etc.). We will not describe all of them but focus on the types of users directly involved in the diagnostic and solution of problems with the systems. We identified the following types of users (see table 1):

- *Operators*: Local operators of the machines or end-users.
- *E1*: First line engineers or local engineers.
- *E2*: Second line engineers.
- *E3*: Third line engineers. This line of engineers consists of several subtypes as a function of their expertise. The users can be characterized according to different variables:

- *Education level*. Operators work directly with the machine and do not require high-level education. The rest of the users are engineers.
- *Expertise*. Operators, E1 and E2 engineers have general knowledge about the system. The level of knowledge increases from Operators to E1 and E2 engineers. E3 engineers have specialized knowledge about the different components of the machines. According to the component in which they are specialized several types of E3 engineers can be defined.
- *Degree of access to specific diagnostic tools, and permission to manipulate things in the system*. As we will see, tools that should be used only by engineers are easily accessible by Operators. Other tools are only accessible in the company that designs the machines.
- *Geographical situation of the users*: inside the Design Company (located in The Netherlands) or at the customer site (see fig. 4). In the Design Company there are E2 and E3 engineers. The users outside the company can be found in different countries of Europe, Asia and USA. Users at customer sites are the Operators and E1 engineers. In Asia and USA there are E2 engineers as well.

The roles we found are:

- *Diagnosing and solving problems*. Whenever a problem with the machine occurs the tasks include detecting the cause of the problem and try to solve it. As we will see, all of the roles considered here will assume this role in different situations.

- *Design, implementation and evaluation of the machine*. E2 and E3 engineers have this role. Different subtypes of E3 engineers are involved in the design of different subsystems. E2 engineers work in the design of the machine as a whole.
- *Installation and maintenance of the machine* in the customer's place by E1.
- *Production* with the machine by local Operators.

Table 1. The users and their associated roles

User	Education	Expertise	Geographical situation	Role
Operator	medium	Whole system	Outside	Production Problem solving
E1:1rst line engineer	university	Whole system	Outside	Maintenance Problem solving
E2:2nd line engineer	University	Whole system	Outside Company	Design Problem solving
E3:3rd line engineer	University	Component	Company	Design Problem solving

Objects in the Current Situation

There are several classes of objects that need to be studied; mainly objects in the user interface and other objects that are located in the Design Company. We will focus in the tools that relate to diagnostic and error-recovery tasks.

The Views in the User Interface

The user's interface contains several views (or windows) with tools to support different tasks. For example, a window offers the possibility to show the values of several variables that can be manipulated to change the status of the machine.

One problem is that the different views are not designed in a consistent way. For example, some windows allow to access to linked information by "clicking" buttons that are in the left side of the screen, others show menus on the top part of the screen, and others require to type commands. As said

before, the re-design of the interface aims to make the design of the windows more consistent.

Another related problem is that the Operators should only use the window that is meant for production with the system (so-called Operator's window). The interface however shows and allows access to all of the windows. We think this is not only unnecessary but also dangerous because it allows operators access to tools that, if they are not used properly, can damage the production with the machines.

Tools to Diagnose and Solve Problems

The current tools are all fragmented and some of them are implemented in the User Interface whereas others not.

In the User Interface:

- *Warning and error messages*. The messages normally specify the problem and how to solve it. The problem is that there are not very many of them.
- *Start up software* provides information about the status of each component of the system at the moment of starting up the machine. This information can help to identify the component responsible of the problem. It should only be used by E1 engineers but Operators use it as well (although it is not in the Operator's window). Another problem is that even E1 engineers have difficulties understanding the information provided by this software.
- *Error logging software* writes all of the errors that occur in the different components of the machine while working with it. The main problem is that too much information about each component is provided and the information is not easy to understand. Another problem is that the resulted file is overwritten every 20.000 lines so if the cause of the problem occurs early in the production process, it is very difficult to detect. Although only E2 and E3 engineers should use it, to win time E1 engineers work frequently with it and some customers even pay software engineers to interpret the information correctly.
- *A tool to detect defects in the product* is installed in the interface and can be used by Operators. The problem is that the possible defects are not easy to see directly and the machine does not inform the user about what or where is the defect located. Consequently, the operators have to look actively for any possible defects. The software does not inform about the cause of the defect in the production neither.

In the company:

- *Historical documents*, manuals and emails showing how a specific problem was solved before, but these are incomplete and badly organized.
- *Trace tools* that simulate, reproduce and trace problems are used in the company while working with the prototypes of the machines. They should only be accessed by E3 engineers because they do not offer warning messages and using them inadequately can damage the machines.

Identifying Problems with GTA

Our task analysis showed that the number of breakdowns and problems related with the use of the machines is high and, consequently the machines have stop frequently. Even though there are some tools that help to solve problems, in general they are limited, fragmented, and difficult to use and understand. Furthermore, currently there are not clear ways to identify the causes, nor clear procedures to solve most of the problems. As a result most of the problems are solved by trial and error. For example, designers involved in the implementation of the machine tend to open the machines to look for the cause of the problem. This solution slows down the design process because it takes around 10 hours to make the machines functional again.

One of the most serious problems we found is related to the access of the tools used to solve problems. The access is not always as established in the norms. As a result users are often tempted to apply procedures they are not supposed to perform and are not trained for. On the other hand, sometimes a procedure cannot be performed because the object cannot be accessed in a situation where it would allow a knowledgeable actor to perform the appropriate task.

Emphasizing Tasks and Users

GTA emphasized the analysis of the tasks and the types of users related to them. The information obtained however, did not seem enough to understand the whole situation and identify all of the aspects needed for the redesign of the system. One of the aspects most difficult to represent with GTA was the differences due to geographical distribution and the transfer of information between different types of users. Consequently, to improve our analysis, we decided to use the DC framework to describe the situation emphasizing these aspects.

Using DC: Multiple Complex Organizations

Apart from analyzing the users and the current tool used to solve problems, we needed to analyze the relationships between different types of users, and between users and tools. Furthermore, we found out that in this complex situation different socio-technical systems needed to be considered:

1. *"Local" socio-technical systems* defined when only one company (Customer Company or Design Company) is involved in the solution of a problem with the machines.
2. *"Distributed" socio-technical systems* when interactions between customer's companies and the design company are needed.

The socio-technical system to be considered depends on the complexity of the problem with the machine (input) and the knowledge resources. Briefly, if the problem was too difficult and could not be solved within a "local" socio-technical system (e.g. a Customer Company), the problem was transferred and the original "local" socio-technical system became a "distributed" one (see below section "*Transmission of Knowledge*"). As explained below, an extra difficulty we had to consider in the analysis was the fact that the distribution of knowledge between the socio-technical systems was different in different continents.

Distributed and Shared Knowledge

The knowledge needed to diagnose and solve problems is mostly distributed among types of users and artifacts. In other words, different types of users have different knowledge about how to solve problems. The same can be said about the current tools used to diagnose and solve the problems (see section "*Tools to Diagnose and Solve Problems*"). The distribution of knowledge between users depends on:

- *Educational level*: Operators versus Engineers.
- *Expertise*: the whole system versus components.
- The *access to tools and permission to manipulate the system* is also distributed. For example, trace tools can only be accessed by engineers working in the Design Company.

However, part of the knowledge is shared among users:

- E1 engineers share the knowledge of the Operators about diagnosing a problem. The difference is that E1 engineers have deeper understanding

about the situation and have also specific knowledge related to maintenance issues.
- E2 engineers have even a deeper knowledge about the machines and how to identify and solve problems with the machines.
- The different types of E3 engineers have some knowledge about the whole systems but they are highly expert in specific components of the machine.

Transmission of Knowledge

The Norms

When the manufactured products have any defect, the Operators can use the software in the interface to detect defects. In general, when the Operators are unable to solve a problem, they should contact E1 engineers. As explained above, E1 engineers have more general knowledge about the system than the Operators and have permission to access tools in the interface that are designed to solve some of the problems. If E1 engineers cannot solve a problem, they are supposed to contact with E2 or E3 engineers. The choice of calling E2 or E3 engineers depends if E1 engineers localized the component of the system with problems or not (see figure 3).

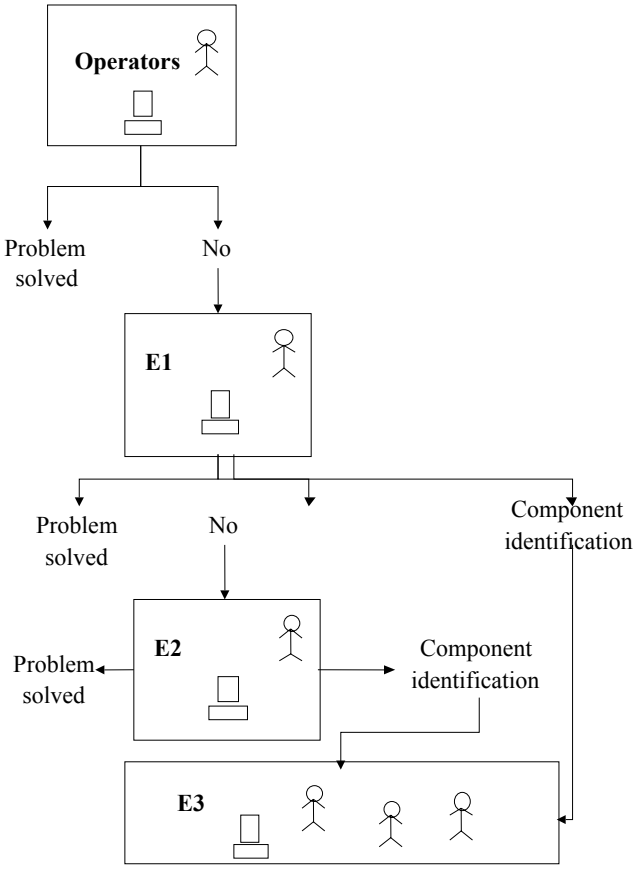

Fig. 3. The transmission of knowledge according to the norms

The Real Practice

The Design Company sets policies on the use of the machine. The policies state the rules to access different tools in the machine and establishes what tasks should be performed for each type of users. The reality however is that customers do not always follow the rules. For example, when the machine stops the Operators should immediately contact E1 engineers. But to save time, they frequently access the "start up software" and try to identify the damaged component. If they are not able to understand the information provided by this software, the Operators end up replacing

some components of the machine trying to find the solution by trial and error. The main problems related to access to knowledge are:

- Start-up software: used by Operators when only E1 should use it.
- Error login software: only for E2 and E3 engineers but Operators and E1 engineers use it. Some Customer Companies even pay extra software engineers to understand error logging software.
- Trace tools: should only be used by E3 engineers but E2 engineers use them.

The Format of Knowledge Transmission

The knowledge is mostly transmitted through verbal language in different forms like, telephone calls, emails, reports, error log and sometimes face-to-face interactions etc.

Factors Affecting the Transmission of Knowledge

When considering the companies in different continents we found out relevant differences that affected the transmission of information. The main factors where: differences in the work organization and differences in the way information is transmitted between users and between users and artifacts.

Work Organization

The work organization in different continents affects the transmission of knowledge (see figure 4):

1. In USA Operators and E1 engineers are in the same organization. E2 engineers are located in USA so they do not have to contact E2 from the Design Company.
2. In Asia E1 and E2 are in the same organization and close to the customers companies.
3. In Europe E1 and Operators are in the same organization. E1 and E2 are in the design company.

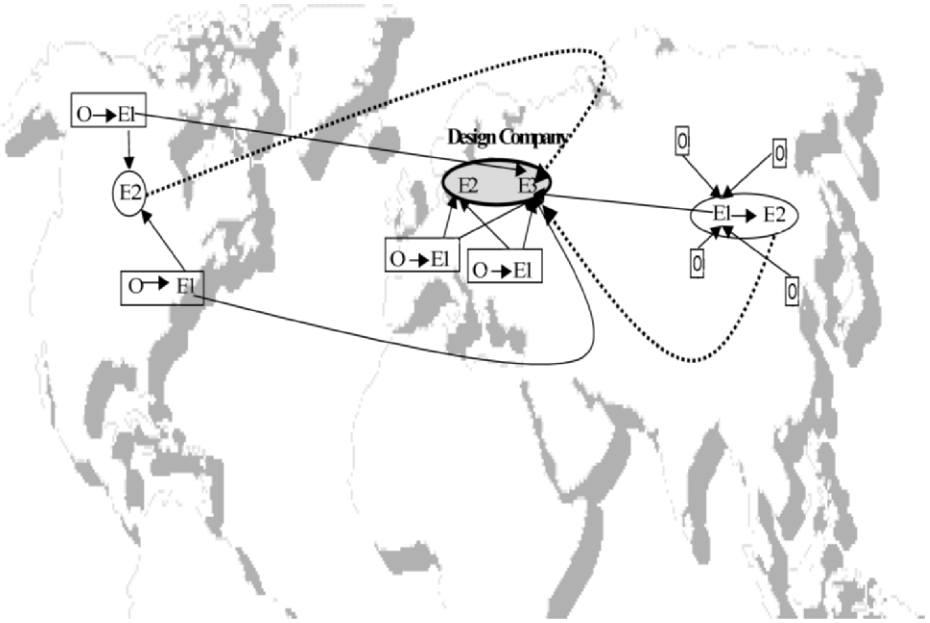

Fig. 4. Transmission of information in different continents

Transmission of Knowledge in Different Socio-Technical Systems

The fact that different roles can belong to different countries was an extra source of problems. To transfer the problem means to communicate with people from other cultures (national as well as professional). In this respect the language proved to be a serious problem. The problems are not only due to the fact that the interface, the manuals and the tools to solve problems are in English. The cross-cultural communication is in English as well and this is a problem especially for Operators and Engineers in Europe and Asia. For example, many complains come from Asia where Operators and Engineers have problems with English, making the use of the systems, including reading the manuals and the communication, more difficult.

The different types of users represent in fact different professional cultures. This situation brings an extra problem specially related to the use of manuals. The manuals for using the machine are written by the E3

engineers who, as any professional culture, have developed a special language that is difficult to be understood by other categories of users.

Conclusions

Modelling task knowledge is not an easy activity, especially when it is necessary to describe and analyze a complex situation. This paper shows how we applied two different frameworks, GTA and DC, to understand and model task knowledge of a very complex interactive system where different types of users, different organizations and cultures needed to be considered. GTA and DC examine the situation from different points of views emphasizing different aspects of the task situation and the information obtained is complementary and necessary when analyzing extremely complex systems.

As a conceptual framework for task analysis, GTA helped us to analyze the situation focusing in the identification of multiple users and the associated roles. The ontology specified by GTA, and the use of EUTERPE to model the task knowledge, proved to be a great help in order to understand clearly the relationships between types of users and roles. Even though the GTA framework was a valuable basis for a first approach to the task situation, the information obtained did not seem enough to understand the whole situation and identify all of the aspects needed for the redesign of the system. Notions such as transmission of knowledge, cultural differences, both regional and professional, needed to be taken into account.

Usually DC has been used to describe rather “simple” settings such as cockpit or software teams [e.g. 3, 6]. In this paper we extend the use of DC to a much larger setting that comprises multiple organizations located in different geographical regions. DC allowed us to understand that, when considering the way information is transmitted and transferred, we were studying a case of multiple organizations. Furthermore, DC offered a clearer framework to analyze what knowledge was shared among users and artifacts, and what aspects were distributed.

Concluding, especially in the case of complex task situations, several points of view need to be considered in order to analyze, describe, and model the task world.

References

1. Chisalita, C., Puerta Melguizo, M.C. & van der Veer, G.C. (2002). Designing Complex systems in Industrial Reality: A study of the DUTCH approach. In I. Aedo, P. Díaz, & Fernández, C. (Eds.). *Interacción 2002: III Congreso Interaccion Persona-Ordenador.* ISBN: 84-607-4501-5.
2. Decortis, F., Noirfalise, S. & Saudelli, B. (2000). Activity theory, cognitive ergonomics and distributed cognition: three views of a transport company. *International Journal of Human-Computer Studies,* vol. 53, issue 1, pp. 5-35.
3. Flor, N.V. & Hutchins, E. (1992). Analyzing Distributed Cognition in Software Teams: A Case Study of Collaborative Programming During Adaptve Software maintenance. In Koenemann-Belliveau, Moher & Robertson (Eds.) Empirical Studies of Programmers: Fourth Workshop, Norwood, NJ: Ablex.
4. Hutchins, E. (1990). The technology of the Team Navigation. In Galegher, Kraut & Edigo (Eds.) Intellectual Teamwork, NJ:LEA.
5. Hutchins, E. (1995). Cognition in the Wild. Massachusetts. MIT Press.
6. Hutchins, E. (1995). How a cockpit remembers its speed. *Cognitive Science*, 2, pp. 14-39.
7. ISO 9241-11 (1998). Ergonomic Requirements for Office Work With visual display terminals (VDT)s – Part 11 guidance on usability.
8. Preece, J., Rogers, Y. & Sharp, H. (2002). Interaction Design, beyond human computer interaction. Wiley & Sons, Inc.
9. Rogers, Y. (1992). Ghosts in the Network: Distributed Troubleshooting in a Shared Working Environment. In Turner & Kraut (Eds.) *Proceedings of the Conference on Computer Supported Cooperative Work*, pp. 346-355, ACM: NY.
10. Rogers, Y (1993). Coordinating computer-mediating work. *Computer Supported Cooperative Work*, 1, pp. 295-315.
11. Rogers, Y. & Ellis, J. (1994). Distributed cognition: An alternative framework for analyzing and explaining collaborative working. *Journal of Information Technology, 9,* 119-128.
12. Van der Veer, G.C. (1990) *Human-Computer Interaction: Learning, Individual Differences, and Design Recommendations*, Ph.D. Dissertation Vrije Universiteit, Amsterdam.
13. Van der Veer, G.C., Van Welie, M. & Chisalita, C. (2002). Introduction to groupware task analysis. In: C. Pribeanu & J. Vanderdonckt (Eds.). *Task Models and diagrams for user interface design.* TAMODIA'2002. INFOREC Printing House, 32-39.

14. Van der Veer, G.C. & Van Welie, M. (2000). Task based GroupWare design: Putting theory into practice. In: *Proceedings of DIS 2000,* New York, United States.
15. Van Welie, M. (2000). Task-Based User Interface Design. SIKS Dissertation Series 2001-6. Vrije Universiteit, Amsterdam. Nl.