

# Groupware Task Analysis in practice: a scientific approach meets security problems

Reinard van Loo, Gerrit C. van der Veer, Martijn van Welie

Dept Computer Science, Vrije Universiteit, De Boelelaan 1081 A,  
1081 HV Amsterdam, The Netherlands. gerrit@cs.vu.nl

## ABSTRACT

Groupware Task Analysis (GTA) is applied for the redesign of a commercial security system. The problem is the confidentiality of the knowledge on the task domain. The system is the actual security systems in use in companies, AND, obviously, none of them is eager to have details of its security management situation and security procedures being made available to outsiders, even if these are employed by a company that designed their system. The paper describes the problems in applying GTA and shows ways to cope with the restrictions involved in analysing safety critical systems in industrial design projects.

**Keywords** Design, groupware, GTA, security systems, task analysis

## GTA - AN APPROACH TOWARDS DESIGN OF COMPLEX SYSTEMS

GTA (Van der Veer, Lenting, and Bergevoet 1996) is an approach to the early phases of the design of complex systems (task analysis, global design) that integrates ethnographic techniques and interaction analysis, analytic techniques from HCI, formal task modelling, and design space analysis. Analysing a complex system means analysing the world in which the system functions, or the "context of use". GTA deals with the context of use of a system in the broadest sense. In traditional literature on task analysis from the HCI (Human-Computer Interaction) mainstream, the focus is mostly on users, tasks, and software. Design approaches for GroupWare and CSCW (Computer Supported Collaborative Work), on the other hand, often focus on analysing the world first of all from the point of view of the (physical and social) environment. GTA, integrates viewpoints from the different approaches.

## GTA Viewpoints on the Task World

1. **Agents** are considered in relation to the task world, hence, we need to make a distinction between actors and the roles they play. Moreover, we need the concept of organisation of agents.

- The label *actors* mostly refers to individual people, either individual or in groups. In situations where modern information technology is applied, actors will sometimes be non-human agents, or systems that comprise collaboration between human agents and machine agents.
- *Roles* indicate classes of actors to whom certain subsets of tasks are allocated, by free choice or as the result of the organisation. By definition roles are generic for the task world. More than one actor may perform the same role, and a single actor may have several roles at the same time.
- *Organisation* refers to the relation between actors and roles in respect to task allocation. The organisation describes the agent structure in the task domain. Delegation and mandating responsibilities from one role to another is part of the organisation, as is the way roles are allocated to actors.

2. We focus on the structural as well as dynamic aspect of **work**, hence, we take 'task' as the basic concept, and 'goal' as an attribute. We make a distinction between tasks (which have a goal) and actions in the 'classical' HCI terminology, and, moreover, we will elaborate task structure and the structure-related concepts of protocol and strategy.

- *Task* can be identified at various levels of complexity. Complex tasks may be split up between actors or roles. Users' unit tasks (and systems' basic tasks) may be decomposed further into (user) *actions* and (system) events, but these can only be understood in a frame of reference created by the corresponding task, i.e., actions derive their meaning from the task.
- The *task structure* will often at least partially be hierarchical. Task structures are not always known by single actors, mainly when different roles are involved in performing different subtasks. On the other hand, performance on certain subtasks may influence the procedures for other subtasks, which makes task flow and data flow structures relevant.

- *Protocols* indicates actual 'rules' as turn out to be applied for decomposing tasks, to be distinguished from 'rules' that may be stated explicitly in instructions which are sometimes not actually followed. Protocols may be situated, i.e., the environment and the presence of actors with certain roles may constitute conditions for protocols to be triggered. *Strategies* indicate structures that can be considered protocols used mainly by experts or typically preferred by them. Strategies will be role related.

3. Analysing a task world from the viewpoint of the **situation** means detecting and describing the environment (physical, conceptual, and social) and the objects in the environment. Object description includes an analysis of the object structure.

- Each thing that is relevant to the work in a certain situation is an *object* in the sense of task analysis. Objects may be physical things, or conceptual (non-material) things like messages, gestures, passwords, stories, or signatures.
- The task *environment* is the current situation for the performance of a certain task. It includes actors with roles, conditions for task performance and for strategies and protocols, relevant objects, and artefacts like information technology that are available for subtask delegation. The *history* and temporal structure of relevant events in the task situation is part of the actual environment. The environment features as condition for task structures (inclusive protocols and strategies as far as these are situated).

### **The first phase in task analysis applying GTA: analysing the current task situation (Task model 1)**

Task analysis may include different activities: (a) analysing a "current" task situation and modelling this (Task model 1), (b) envisioning a task situation for which information technology is to be designed (Task model 2), and (c) specifying the semantics of the information technology to be designed. Our current experience concerns phase (a) only, so we will not go into details of the other phases of task analysis activities.

In many cases the design of a new system 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 and modelling of the current situation may help formulate design requirements, and at the same time may later on allow evaluation of the design. Collecting task knowledge for task model 1 will need to be done by applying a variety of techniques. These techniques may be derived from "classical" HCI design methods like TKS (Johnson, Johnson, Waddington, and Shouls 1988) and MAD (Scapin and Pierret-Golbreich 1989). Often we will also need hermeneutic interpretation techniques (Van der Veer 1990). Additionally we will need to adopt ethnographic techniques that have been used for groupware design and CSCW, like interaction analysis (Jordan 1996). Actual techniques to collect the raw data for GTA normally include ethnographic observation, camera recording of interactions, and tape recordings of interviews

### **EUTERPE, a Task Analysis Tool**

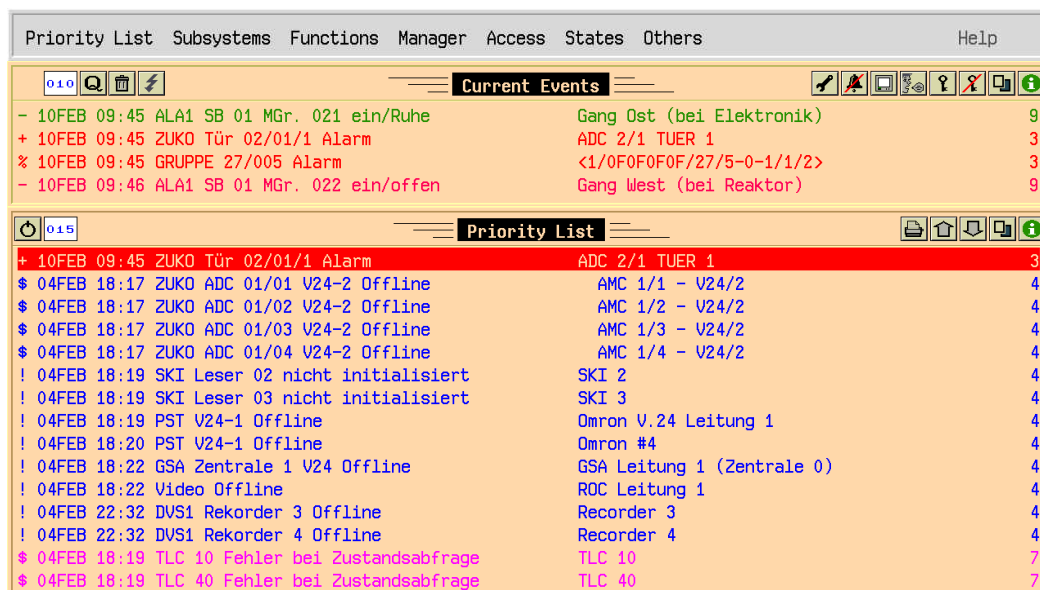
Our task analysis environment EUTERPE (van Welie, van der Veer, and Eliëns 1998) has been developed to aid the process of task analysis in GTA. EUTERPE was developed because task analysis is still an activity that needs support. After the task world has been modelled it is up to the analysts to interpret the task model and find out where causes of problems can be found or where there is room for optimisation of the work. A task model that can describe the task world including co-operative aspects and that allows some form of analysis could improve the task analysis process and outcome. Preferably the analysis of the task model should be done (semi-) automatically, thereby reducing the required effort of the analysts. However, performing a formal analysis of a task model requires a formal representation of the task model that is suitable for doing an analysis, especially for analysing co-operation. The task model therefore needs to be based on a task analysis theory that recognises the co-operative aspects of the task world. Although a formal analysis can be the basis for analysis it is not on the level analysts prefer to work. Hence representation tools can effectively hide the formalism and provide means to assist in analysing the environment that is being studied. In addition, a tool can also provide more structured ways of doing task analysis. EUTERPE, based on GTA, supports formal analysis both on a logical and a visual level. It uses formal representations internally while offering graphical representations to the user.

### **AN INDUSTRIAL DESIGN CASE: THE SECURITY SYSTEM**

Currently the GTA-Method is applied at the Austrian Research Centre Seibersdorf (ARCS) mainly for the redesign of a security system marketed by Philips Industry, Austria. The main problem in this case is related to the confidentiality of the knowledge on the task domain. It is the actual security systems in use in these companies that is the basis for our knowledge of the task domain. Obviously, none of these companies is eager to have details of its security management situation and security procedures being made available to outsiders,

even if they are employed by a company that designed their system. This paper describes the problems encountered in applying GTA in a real life redesign situation and shows ways to cope with the restrictions involved in analysing safety critical systems. Additionally, this paper derives additional suggestions on how to adjust task analysis for this type of industrial design projects.

Securing large objects like factories, museums, banks or airfields is no small feat. Monitoring and controlling areas in these objects is done with the help of movement-detection systems, video camera systems, access control systems, fire detector systems, elevator control systems etc. In practice, all the information from these (sub-)systems is led to a control room where human operators have to respond appropriately to (alarm)signals. Keeping an overview on the building status becomes almost unmanageable in complex combinations of sub-systems. To support the operator in monitoring the state of the secured object and to integrate the different subsystems into one system, the sCalable Security System (CSS) has been developed. The CSS integrates the information flow from -and to- all subsystems in one central computer system with a generic (graphical) user interface (see figure 1) available on several workstations.



**Figure 1:** The CSS user interface (menu bar, current events list and priority list)

The general concept of the CSS Operator interface is, that it reports every sub-system message that is defined by the manager as 'relevant to monitor'. Examples of such messages are 'Door contact XXX at ZZZ Alarm', 'Fire detector YYY at ZZZ Alarm', 'Door contact YYY at BBB Malfunction', 'Intrusion subsystem YYY offline' etc. When a message enters the CSS a short beep is generated. The interface shows two lists: one first in-first out *Current Events* list where all relevant messages are listed and a *Priority* list. The messages in the *Current Events* list are supposed to be acknowledged by the operator one-by-one in the order of them entering the system. The list also contains 'information'-type messages (like 'Fire subsystem XXX online') and messages that are important to remember. After acknowledging these last kind of messages they are automatically transferred to the *Priority* list, where they only disappear after the situation causing the message is somehow resolved.

Little knowledge was available about how the current system is used and what kind of problems the users have with the system and/or User Interface in performing their work. To gain more insight on this topic, the first phase of the Groupware Task Analysis (GTA) method was performed. The analysis focussed on the use of the system by the end-users (i.e. operators, porters and system-managers) in their actual use-environments (factory, chemical factory and office buildings were visited). Goal of the analysis was to gain insight in the current use of the system and to propose directions of change/extensions to improve the (practical) usability of the system. The GTA framework, method and tools were highly usable for this.

## TASK MODEL 1 - ACCESSING SOURCES OF TASK KNOWLEDGE ON THE CURRENT SYSTEM

Participating observations and semi-directed interviews were first carried out at two sites that seemed relevant. Getting the necessary co-operation of the managers and employees of the visited sites took some effort. Explaining the goal of the visits by telling some characteristic cases and ensuring that the resulting information remained confidential proved to be helpful in this process. At the end of the visits most managers and employees were enthusiastic about the fact that finally someone took the time to listen to them, and took their grievances serious. Due to the sensitive nature of the work observed (Security control rooms inside the objects to be secured) camera recordings (as usual in observations) and tape recordings of interviews were (almost) not possible. This did not impede the data gathering process.

Crucial to the acceptance of GTA in the project-organisation were the following factors:

- **External funding.** The analysis of the CSS was carried out in the context of the European OLOS Project (EC Human Capital Programme CHRX-CT94-0577). This made it possible for ARCS to get to know GTA with low financial risks. The results were the main ground for extending the appointment of the researcher at ARCS, outside European funding, to apply the method also on other projects.
- **Complementary expertise.** The results of the analysis made visible a gap in the acquired expertise in the project organisation: The direct translation of context-of-use characteristics in design information.
- **GTA being a method.** It offers a more solid base to work and communicate from. One does not start from scratch. Also, in getting support for gathering your data it is also very handy to mention that you are using a method. In a way, one abuses connotations of the method-concept like 'objectivity' and 'being systematic'; notions that make a technology oriented organisation more receptive

In successfully accessing the sources of task knowledge the following factors were crucial:

- **Confidentiality.** When performing several levels of confidentiality have to be kept. As with ethnographic observations in general, by being an observant on site one enters a social setting with its own rules, hierarchy etc. For the observed and interviewed operators it is important that what is observed does not impede the relation with management. Depending on the situation one stresses this point more (where a tension between management and operators is present) or less (where management and operators have a more open working environment). For the managers it is more important that the results of the analysis are treated with care and remain confidential. During the visits all these concerns could be resolved.
- **Not directly evaluating people.** On a more personal level, assuring that it is not a evaluation of the performance of the operators or managers in performing their job makes the people present in the use-organisation more co-operative.
- **Making clear Focus analysis.** The goal (improvement of the CSS) and focus (People performing work in an environment) of visits at use-organisations have to be made clear to all participants. This focus in connection with improving system design has to be explained further because normally these two are not considered in connection with each other. A characteristic case where it is clear that the system was not attuned to the use-organisation on the work-level can help (see for instance, (Neumann 1994)). It opens up a way of thinking about work and the role of the system in it where the participants can easily relate to.

## PROBLEMS OF COLLECTING AND MODELING TASK KNOWLEDGE OF SECURITY SYSTEMS

First of all we had to choose which sites should be visited for our data collection. The following criteria were used in selecting sites:

- **Type of secured object.** The CSS security system is in use at over 70 sites including banks and factories. Every installation is tailored to those aspects that are most important to maintain or observe: At banks, for instance, maintaining control over who entered the building, and the monitoring of some secure areas are the primary concerns. At Factories, however, the primary goal is mostly the prevention of fire. These differences are also reflected in the work performed in control rooms.
- **Number of subsystems used.** The number and kind of subsystems differ considerably between sites: From simple (2 subsystems) to very complex (10-15 subsystems, 60.000 detectors etc.). This complexity also has its consequences for the work being performed.
- **How long in use.** The time the CSS has been installed also differs per site. At some sites, the system is rather new to the operators while at other sites the operators have accustomed themselves to the system. This also has consequences.
- **In flux or not.** Some sites have a very stable configuration, at other sites constant changes are being made to the configuration: new subsystems are added, construction work changes the layout of the building requiring the replacement of cameras, detectors and the like.

- Due to the limited time, we choose to start with as much **complexity** as possible. Two different kind of secured objects (office building and factory) were visited. Each site had many subsystems, the CSS had been installed for some time, and the configuration was in flux -so that the operators were able to compare the old with the new situation. Later on two other sites with low complexity and low flux were paid a short visit to verify and to get some feedback on observations done at the first two sites.

Using the chosen task knowledge gathering techniques (Ethnographic observation and semi-structure Interviews) in a straightforward way was severely impeded by situational and practical circumstances. Therefore, a more dynamic/improvisational mode of use of these techniques was taken. Videotaping during the observations was not allowed for security reasons. Therefore a 'paper-and-pencil' method was adopted for 'recording' the observed work. During the observations this was not perceived as an impediment to the analysis. It does however have consequences for how one works. Usually, video recordings are, among other things, used to analyse specific interesting actions: Interaction analysis. Not being able to do this afterwards calls for interaction analysis *on the spot*. A positive side effect of this is, that immediate feedback can be obtained from the observed persons. The cycle then becomes one of *observe – analyse – feedback*: One observes an interesting event (while not known yet, or along the way identified as a *hot spot*, see (Jordan 1996)), writes down a short task structure, and then the observed person can be asked about the event. This gives the possibility to check the interpretation just made. This also calls for another mode of observing, more active on the observer part: more intrusive. It is noted that the observed people do not mind this, as long as the observer stays close to the work. Tape-recording the interviews gave fewer problems. At most of the sites it was allowed to use the tape recorder to record the interview. At one site it was, however, not allowed. Not being able to record an interview was perceived as a severe impediment. On top of this, the interviews with operators at that specific site were only allowed in the presence of the security-manager, making the interviewed person less open about their work. Also, the manager only had the time to be present at one interview, so only one interview in total could be held. However, remaining questions could be resolved during a later short visit to the office building.

The interviews were done using a semi-structured approach, the so-called 'What-How-Why' role-specific interview as proposed in (Sebillotte 1995). This is used to gain insight into the task structure as explicitly seen by an actor. Using this method the problem of 'talkative' interviewees emerged: The semi-structured interview method had different results with different people. The interviewed operators were very short and not precise in their answers, leaving little room for the interviewer to prod deeper. During the observations, however, the same operators had much less difficulties in talking about their work. Whenever something happened and one was asked about it detailed, general, descriptions could be given. The interviewed managers, however, could describe their work in more detail. It is felt that the type of person and the situation in which the person was interviewed had a great influence on the quality and depth of the interview.

During task modelling (using EUTERPE) it was found that the **interviews** resulted in more *complete* information about the work being carried out. The overall structure and personal interpretation of the work became clearer. The **observations**, however, resulted in more *detailed* and *correct* information about the work being carried out. In the task analysis these prove to be highly complementary to each other: correcting, filling up the blank spaces and giving structure to observed actions.

## ADAPTING THE GTA APPROACH AND TECHNIQUES

Depending on the situation, different data gathering techniques are needed. What GTA lacks at this moment is a 'toolbox' of data gathering techniques with which one can start an analysis. These tools should act as building blocks or inspiration for new data gathering processes, where the situational constraints determine the choice of technique or combination of techniques. Such a toolbox should describe several techniques and, more importantly, their characteristics should be made explicit. What is it that the technique actually does? What kind of data does it produce? In the PD and HCI fields many different techniques can be found which are regarded as useful for performing task analyses. The criteria of application of such techniques in our approach should be that the results of the technique add to a better understanding of the work in terms of the GTA framework.

The analysis as performed is the first step in the GTA method and was directly aimed at a kind of 'evaluation' of the studied system. This influences task modelling and the communication of results. One influence was to have the surface structure of models made to incorporate aspects that had to be made explicit for communication or for understanding by the analyst himself. An example was the use of new constructors: The usual task constructors ALT, PAR, SEQ etc. were added with 'conditional' versions of it: CALT, CPAR etc. This was

meant to make clear that the execution of the tasks could be interrupted by a higher priority task, after which the old task was picked up again. This was such an important aspect of the observed work that the surface of the task model should signal this.

During analysis the GTA *framework* proved to be the greatest help. It worked as a kind of 'check-list' to focus attention to things that matter in performing tasks. During analysis it was felt that *methodical* support was lacking, though there seemed not to be a real need for this. More needed is an openness on the side of the analyst in learning to understand why people do something. One has to be able to '*just enter a room*' and start observing something, which might turn out useless or useful for the analysis. One has to be able to deal with the open-endedness associated with a more 'ethnographic' style of performing analysis. That the GTA method leaves enough room for this without losing the focus for system design as witnessed in the framework and later stages is regarded as a strong point, not as a weak.

## CONCLUSIONS - GTA IN INDUSTRIAL PRACTICE

The promise of GTA as a system design method could not be validated in the case of our field investigations. In this project our help was called for late in the system design process and changing the process in terms of the GTA design approach was not possible. This is, however, not considered as a problem: as part of a larger design team and project culture the results of the analysis stated in the GTA framework (work, people, Environment) and the translation of these findings into system design are incorporated in the project. Especially the 'translation' is crucial, and this turned out to be part of the core benefit of the GTA method in our case. The outcomes of the analysis were translated in concrete recommendations for interface and system design. This way the results were regarded very beneficial to the overall project. One acts as an intermediary between users and system designers and one has to be able to speak both languages. If one stays too far away from either the conceptual world of the users or of the designers the method won't work.

Reactions to the GTA method are very positive and money is spent to carry on the work within the GTA framework. This does not mean that the application of GTA is without problems: What lacks is more 'awareness' in the company of what it means to perform task analysis's at the customer and how it should be integrated in the already present system design culture. To the majority of traditional IT-personal and organisations it is radically new. It takes time to let this awareness grow by (partly) applying the method in current projects. Time is needed to move the application of the method forward in new projects.

## REFERENCES

- Johnson, P., Johnson, H., Waddington, R. and Shouls, A. (1988), *Task-Related Knowledge Structures: Analysis, Modeling and Application*, in: Jones, D. M. and Winder, R., *People and Computers IV* pp. 35-62, University Press, Cambridge.
- Jordan, B. (1996), *Ethnographic Workplace Studies and CSCW*, in: D. Shapiro, M. J. Tauber and R. Traunmueller, *The Design of Computer Supported Cooperative Work and Groupware Systems*, North-Holland, Amsterdam.
- Neumann, P. G. (1994), *Computer-Related Risks*, Addison-Wesley, New York
- Scapin, D. and Pierret-Golbreich, C. (1989), *Towards a method for Task Description: MAD*, in: Berlinguet, L. and Berthelette, D., *Work With Display Units 89*, Elsevier, Amsterdam.
- Sebillotte, S. (1995), Methodology Guide to Task Analysis With the Goal of Extracting Relevant Characteristics for Human-Computer Interfaces, *International Journal of Human-Computer Interaction*, vol 7, no. 4, pp.341-363.
- Van der Veer, G.C. (1990) *Human-Computer Interaction: Learning, Individual Differences, and Design Recommendations*, Ph.D. Dissertation Vrije Universiteit, Amsterdam.
- Van der Veer, G.C., Lenting, B.F. and Bergevoet, B.A.J. (1996), GTA: Groupware Task Analysis - Modeling Complexity, *Acta Psychologica*, no. 91, pp.297-322.
- Van der Veer, G. C., van Vliet, J. C., and Lenting, B. F. (1995), Designing complex systems - a structured activity, *DIS'95, Symposium on Designing Interactive Systems*, ACM Press, New York.
- van Welie, M., van der Veer, G. C., and Eliëns, A. (1998), Euterpe - Tool support for analyzing cooperative environments, *Ninth European Conference on Cognitive Ergonomics*, Limerick, Ireland.