Modeling Complex Processes in GTA

G.C. van der Veer, M. van Welie, D. Thorborg[†]
Vrije Universiteit, Department of Computer Science
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
[†]Linköping University, Department of Computer and Information Science,
581 83 Linköping, Sweden, email: {gerrit,martijn,thorborg}@cs.vu.nl

ABSTRACT

GTA is a method for GroupWare task analysis. Representations of task models in GTA originally focused on object oriented templates of the various elements, and on hierarchical relations between elements as seen from separate viewpoints (work, agents, and situation). In order to model time aspects and complex processes, we introduce new concepts as well as new types of representation inspired by workflow representation techniques.

Keywords

GTA, process models, task knowledge, task modeling, workflow

ANALYTIC MODELING IN DESIGN

Systematic design of complex systems starts with a phase where the current work situation is analyzed. Knowledge of a current work situation, collected by knowledge elicitation, ethnography and other methods, is of a broad variety. An important step in this phase of design is to model all relevant task knowledge, including both static knowledge and process knowledge, in a model that includes indications of conflicts, diversity in procedures and situational conditions of tasks. The resulting representation of this phase of task analysis we will label “Task model 1” (van der Veer et al. 1995) (TM1). TM1 is valid if it represents all relevant aspects of a current work situation as far as the knowledge is needed for future (re)design.

Ideally, the next step in design is to envision the new world, i.e., the work situation for the case that the intended design is finalized and the specified technology is implemented. New technology will change the work situation, and will often strongly impact the work organization and procedures. All of this is part of the envisioning, so all of this should be represented in this phase of design. We speak of “Task model 2” (TM2) to indicate a representation of the envisioned new task situation. TM2 should represent the same aspects of work, work organization, and work situation, in order to allow analysis of the changes and to make decisions for detailed design.

The development of TM2 will be based on TM1 (including the conflicts, needs, and diversities recorded), on negotiation with the client of the design (including standards, corporate image, usability criteria, budget for design and implementation), and on status and envisioning of enabling technology (situational constraints, predicted possible developments). In later phases of design (specification, evaluation) detail specifications will often require a re-consideration of TM2, and, hence, new negotiations with the client of design.

Many HCI task analysis methods cover both phases of task analysis as well as detailed design (Johnson et al., 1988; Johnson, 1992). On the other hand, some authors do not indicate the stages their approach is meant for: GOMS (Goals, Operations, Methods, and Selection rules: Card, Moran, and Newell, 1984) can be applied for any of them or a combination.

Analysis of the current task situation: Task model 1

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 of the current situation may help formulate design requirements, and at the same time may later on allow evaluation of the design. In all cases where a ‘current’ version of the task situation exists, it pays off to model this.

Sebillotte (1988) presents a method to collect task knowledge and structure this into a hierarchical model of subtasks. Scapin and Pierret-Golbreich (1989) elaborate on this method and provide an object-oriented formalism for modeling knowledge of existing task situations, like Sebillotte mainly focusing on activities. The main method used by Sebillotte and Scapin, e.g. Methode Analytique de Description (MAD) is a structured interview, focusing explicitly on the hierarchical relation of tasks. MAD specifies the decomposition of a task into subtasks, with ‘constructors’ that indicate the conditional and temporal decomposition, e.g., OR (subtasks are alternatives), COND (subtasks performed only if a condition is valid), LOOP (subtasks performed repeatedly), PAR (subtasks performed in parallel). Constructors can be combined (COND.LOOP indicates that under certain conditions the following loop is performed). Constructors of this kind represent the task structure as elicited by the interviews. In different task domains a slightly different set of constructors may be found. Interestingly, methods like MAD even offer a completely object oriented method for representing alternative ways of decomposing a task (Scapin and Pierret-Golbreich, 1989).

The same type of task information features in other HCI task analysis methods, e.g., Task Knowledge Structures (Johnson et al., 1988). Taxonomic substructures in TKS represent subtask structures as well as knowledge about objects (in the sense of ‘things’) and their semantic relationship. This knowledge can be seen as declarative. The objects are represented in an object-oriented formalism.

The various methods of task analysis in HCI and the related formalisms have much in common: a mostly hierarchical model of task structure, representing task knowledge of actors in the task domain. This model is in most cases completed by both procedural and declarative task knowledge related to, respectively, task decomposition and task related knowledge of the semantics of objects (Diaper, 1989). Task models of this type are intended to describe the situation as it can be found in real life, by asking or observing people who know the situation from experience (e.g. Johnson, 1989). Task model 1 is often considered of a generic nature (e.g. Sebillotte), indicating the belief of authors in this field that different expert users have at their disposal basically the same task knowledge.

Specification of the future task situation: Task model 2

Many design methods in HCI that start with task modeling are structured in a number of phases. After describing a current situation (task model 1) the method requires a re-design of the task structure in order to include technological solutions for problems and technological answers to requirements. Johnson et al. (1988) provide an example of a systematic approach where a second task model is explicitly defined in the course of design decisions. The design decisions that lead from task model 1 to task model 2 will in actual situations be based on three different sources, which each provide specifications for a new situation.

- Problem analysis: The actual system, as modeled in task model 1, will often be considered (by users and stakeholders, i.e., the spokesmen that provide the input for the first task model) far from ideal. Their knowledge and attitudes, as far as made explicit during the first phase of task modeling, will help the designer to specify parts of the task structure, and characteristics of task related objects, that require a change to optimize the task performance. In our design practice we discovered that classical HCI methods for collecting task knowledge will only reveal part of the problems, though. Ethnography (see below) may provide complementary specifications. Combining the various techniques of collecting knowledge about the task domain, and modeling the total, allows us to identify problems, conflicts, and inconsistencies in the current situation.
- Client's specifications: The requirements as stated by the ‘client’ (the instance that ‘pays’ the design team for improving the task situation) have to be clearly distinguished from the users' problem specifications. Clients (although they will sometimes also be actors in the task domain and users of the technology to be considered) may provide completely different aspects for design, like economic considerations, time constraints, and usability specifications for acceptance testing. In as far as clients' specifications are in conflict with proposed solutions to problem specifications, the designers will have to negotiate with the client.
- Technical specifications: Technology plays an important role in the definition of specifications. On one hand it serves as a base for the designers' insight needed to meet specifications from the previous sources i.e. what technological possibilities can serve a solution. On the other hand technology itself defines requirements (constraints) i.e. what solutions are feasible. Design decisions related to the application of new technology, the introduction of new artifacts and the definition of new work procedures will consider technical feasibility as well as the specifications derived from knowledge of the actors and from requirements of the client.

Providing this structure of sources of design specifications does not automatically lead to task model 2, and does not guarantee optimal design decisions. It merely serves as a base for structuring problems, alternative solutions, and criteria, and, hence, can be helpful in developing an explicit design rational. In the current paper we will not elaborate on the process of developing task model 2. For an elaborate overview of different problems related to bridging the gap between task model 1 and task model 2, and for illustrations of solutions as applied in actual situations, see Wood (1998).

Task model 2 will in general be formulated and structured in the same way as the previous model. However, is not considered a descriptive model of users' knowledge, although in some cases it might be applied as a prescriptive model for the knowledge an expert user of the new technology should possess.

DESIGN APPROACHES FOR CSCW AND GROUPWARE

CSCW (Computer Supported Collaborative Work) work stresses the importance of group phenomena and organizational structure and procedures. Most CSCW approaches are at least partly rooted in Activity Theory or related theories (Nardi, 1995), as much as most HCI approaches are related to Cognitive Psychology. Activity Theory stresses the importance of the interrelations between actors, objectives, and the community in the effects of using tools, applying rules, and the division of work. Hence, in analyzing complex activities (a concept that can be considered equivalent to the concept of task or goal in HCI task analysis), the 'classical' HCI methods are considered insufficient, focusing only on activities and knowledge of individual actors. CSCW literature strongly proposes ethnographic methods (Shapiro, Tauber, Traunmueller, 1996).

Ethnography literally means describing culture. When ethnographers are engaged in the design of complex technology ('GroupWare' or 'cooperation technology') they study a task domain (or 'community of practice'). They will become a participant observer, if possible with the status of an apprentice, being accepted as an outsider in this respect and being themselves aware of their status of analyzing observer. The ethnographer observes the world 'through the spectacles of the aboriginal' and at the same time is aware of his status of an outside observer whose final goal is to understand and describe for a certain purpose and a certain audience (in the case of CSCW: a design project). Ethnography asks for interpretation from the analyst to a much larger extent than HCI data collection methods do. The analyst (ethnographer) has to choose a focus (or, if needed, more than one) on either activities, environments, people, or objects. The choice of focus is itself based on prior ethnographic observations, which illustrates the bootstrapping character of this type of analysis. Knowledge of individual workers in the task domain may be collected as far as it seems to be relevant, but it is in no case a priori considered the main source, and will never be considered indicative for generic task knowledge.

Ethnography often helps discovering conflicting goals by regarding the history of a community of practice that leads to the current situation. The ethnographic approach is unique in its attention to all relevant phenomena in the task domain that are not explicitly verbalizable by (all) experts.

Ethnographers apply different methods. In our experience with applying ethnography in design of technology for complex task domains one method turned out to be both reliable (i.e., leading to reproducible results in the description and interpretation of task phenomena) and relatively easy to teach to designers of technology: 'interaction analysis' (Jordan, 1996). In this method the analysts, like in all ethnographic approaches, start their observation purposely without a conceptual framework regarding characteristics of task knowledge. Systematic data collection in interaction analysis starts with video recording of relevant phenomena (the relevance of which can only be inferred from prior observation) followed by systematic transaction analysis, where inter-observer agreement serves to improve reliability of interpretation. Interaction analysis covers the methods for information collection and subsequent interpretation that might serve as a basis for developing task model 1 (and no more than this since this specific method only covers information on the 'current' state of a task domain). However, the methodology for the structuring of the collected data and interpretations into a total task domain description is often rather special and difficult to follow in detail.

The general impression is that CSCW design methods skip the explicit construction of task models 1 and 2. After collecting sufficient information on the community of practice (and its history), detailed specifications of the new system are developed, based on deep knowledge of the current task situation that is not formalized. This might cause two types of problems: on the one hand, the relation between specifications for design and analysis of the current task world might depend more on intuition than on systematic design decisions; on the other hand, skipping task model 2 may lead to conservatism in view on organizational and structural aspects of the work for which a system is to be (re)designed.

THE COMPLEXITY OF TASK KNOWLEDGE

In complex situations of people at work, possibly using artifacts, not all phenomena that are relevant for task description are of the same type. For instance not all might be available as verbalizable knowledge of individuals, and not all may be reliably perceived by a trained observer or documented. Hirschheim and Klein (1989) elaborate two dimensions of task knowledge (world knowledge in their terminology): subjectivist-objectivist and order-conflict. The order-conflict dimension stresses the fact that the analyst is not a passive outside observer, but active in shaping the task model. If different actors in the task world hold different beliefs or conflicting knowledge and goals this should be explicitly reflected in the task model 1 and be considered in specifying task model 2 where conflict might be either resolved by forcing the new task situation into a unified solution of the conflict, or, alternatively, by providing multiple possibilities for procedures and private situations as part of the general task world for individuals with conflicting interests. The subjectivist-objectivist dimension of Hirschheim and Klein indicates the problem of collection of knowledge for task model 1. In this respect, we refer to a framework by Jordan (1996), see figure 1.

Relevant task domain information may have to be collected focusing on different phenomena, using different methods of data collection. Based on an analysis of the character of the knowledge sources in this framework, different methods are identified to collect all information needed to construct task model 1. For task knowledge in cell a, psychological methods may be used including those elaborated by Johnson (1989), Sebillotte (1988), and Scapin and Pierret-Golbreich 1989): interviews, questionnaires, think-aloud protocols, and (single person oriented) observations. For knowledge indicated in cell b observations of task behavior will have to be complemented by hermeneutic methods to interpret mental representations (Van der Veer, 1990). For the knowledge referred to in cell c the obvious methods concern the study of artifacts like documents and archives. In fact all these methods are standard to analytic modeling in design.

task world knowledge	**individual**	**group**
explicit	**a**. knowledge and skills	**c**. models/stories/instructions
implicit	**b**. intuition/expertise	**d**. culture/community of practice

Figure 1: Dimensions of knowledge of complex task domains

The knowledge indicated in cell d is unique in that it requires ethnographic methods like interaction analysis. Moreover, this knowledge can be in conflict with what can be learned from the other sources, as is already shown in the examples presented in the previous sections. First of all, explicit individual knowledge often turns out to be abstract in respect to observable behavior, and turns out to ignore the situatedness of task behavior. Secondly, explicit group 'knowledge' (e.g., expressed in official rules and time schedules) often is in conflict with actual group behavior, and for good reasons. In fact, official procedures do not always work in practice and the literal application of them is sometimes used as a political weapon in labor conflicts as a legal alternative for strike. In all cases of discrepancy between sources of task knowledge, ethnographic methods will reveal unique and relevant additional information that has to be explicitly represented in task model 1.

The allocation of methods to knowledge sources should not be taken too strictly. In fact the knowledge sources often cannot be located completely in single cells of the conceptual map. The main conclusion is that we need these different methods in a complementary sense, as far as we need information from the different knowledge sources.

GTA - GROUPWARE TASK ANALYSIS

In designing complex systems, we need to attend to all relevant types of task knowledge. Hence, we need to combine both the different techniques to collect information on the task world, and we need to develop a modeling approach that integrates the relevant concepts that are considered in methods from both the analytical task analysis domain, and from ethnographic methods. GTA (van der Veer et al, 1996a) is a method for task analysis that provides a conceptual framework and related techniques and tools for the analysis and design of complex systems. These systems are often characterized by different users with different roles and competencies and stakeholders (significant others who do not physically use the technology concerned but who's work is affected by it), and by complex technology in complex work situations and work organization. The label Groupware indicates the basic assumption of GTA that complex interactive systems and work processes can not fruitfully be analyzed from the point of view of single work places or single user-computer interactions. Each element of work is part of a large process that involves organizational aspects as well as the situation - both in its spatial relations and in its history of past events. In comparing the different analytic approaches and interaction analysis, we found that the various techniques are based on different viewpoints on the task domain. Analytic methods focus on work and task aspects in all cases, and, moreover, approaches like TKS (Johnson, 1989), focus on roles. A specific exception to the analytic approaches is the work of Walsh (1989) who focuses on objects and related environment aspects in the first place and to work aspects only in the second place. Ethnographic approaches focus in all cases on the situational aspects, although never completely exclusively, other aspects are attended as far as the bootstrapping techniques show their relevance in any actual situation. GTA merges these different points of focus, and, consequently, allows three views on knowledge about the task domain: *work, agents* and *situations*. Together these views provide the conceptual framework for analyzing and modeling TM1 and TM2.

The three perspectives on task knowledge are not independent. Analyzing the task world from the point of view of work, e.g., can only be done fruitfully if one considers the relation between work concepts (tasks, goals) and aspects of the agents who perform tasks, and relations to situational conditions of performing tasks. For each of the perspectives the relation to the concepts of the other perspectives have to be taken into account, i.e., have to be modeled and analyzed. In fact, for each perspective separately this is not new. The main contribution of GTA is to fully stress the need for applying all three of the perspectives, accepting redundancy in favor of the multiple viewpoints.

Work

Modeling work means representing work activities in their relation to each other, as well as to aspects of the other views. The classical approach in this view is best represented in “Hierarchical Task Analysis” (HTA, see Kirwan and Ainsworth, 1992), where work is described as a hierarchical structure of tasks and subtasks. HTA, like other analytic approaches, relates task descriptions to situational aspects. TKS (Johnson, 1992) relates the work view to aspects of agents and organization (especially the "role" concept). Approaches like GOMS (Card, Moran, and Newell, 1983) connect a goal to each task (hence there is a structure of sub-goals). GOMS defines the smallest elements of a task / goal hierarchy as “unit task” (which is the end-node of the task tree, and, as such, from the point of view of the user, the smallest element that is a meaningful complete task with a goal). Tauber (1990), in addition, defines the concept of the system’s “basic task”, indicating a unit of task delegation as provided by technology. The elements of GTA's conceptual framework for the work view consist of a small number of concepts:

- Work may be structured in one or more *task structures*, where the structure of subtasks has to be described by constructors, that are specific to the task domain. Constructors may indicate a temporal sequence, triggering of sub-tasks by each other, loops, conditional choices, etc.
- At the high end of a task structure are *business goals* and tasks related to them. This type of task is often not delegated to a single person or role, and, especially for high level goals there may be several tasks that could provide a way to achieve the goal requiring a flexible structure.
- At the low end of a task structure we find actions, elements that derive their meaning from the unit task they are part of. E.g. hitting a return key is meaningful, but the meaning varies between situations where one is typing text in a word processing, navigating between cells in a spreadsheet, or issuing a command to an operating system.
- A set of tasks that belongs to a single role, i.e., is intended to be performed by a single person, often has a well-defined structure. If the structure is generally considered to be “good” or common work practice, we label this *“protocol”*. If the procedure is characteristic for experts in a certain task domain (e.g., developed on the base of learning or experience) we call this *“strategy”*.

Agents

Many task analysis approaches seem to take for granted that task knowledge of experts contains a considerable part of generic task knowledge. However, in practice we find that complex processes are performed by several people who play different roles. Approaches like TKS (Johnson, 1992) introduce this concept and relate it to a specific set of tasks and a task structure. Most of the classical task analysis techniques only work if we focus on a single role. Ethnography, on the other hand, is strong in investigating whether roles exist and how they relate to structures of work and to situations. In our viewpoint an agent is an active entity, usually a human or a group of humans, but intelligent systems are also considered agents. In analyzing the viewpoint of agents, GTA needs several concepts:

- Groups of *agents* need to be described in terms of their relevant characteristics. This refers to the importance of relating actors to the requirements of tasks. General characteristics that may be relevant depending on the technology applied may be typing skill, experience in using devices like mouse, language abilities. More “remote” variables like spatial ability and fear of failure may impact the interaction between actors and technology in certain situations. Task related aspects of actors refer to knowledge and experience with the domain of work.
- People can take one or more *roles*. Groups of people may in some task domains do the same (the “secretariat” is an instance that may consist of one or more agents, from the outside it performs a single role). Agents may perform several roles simultaneously. The relation between agents and roles may be established and disconnected over time, based on conditions of the situation (election for a post, being the next in the row to succession, self-selection).
- Agents and roles are related to each other in an *organization*. The analysis and representation of an organization should include information on *responsibilities* for tasks, task *delegation*, and the *assignment* of roles. Another part of the analysis of the organization concerns competences and ownership of elements in the situation, like “things” (See below).

Situation

Describing the situation as far as relevant for task analysis is often neglected in classical task analysis. An exception is Walsh (1989) who actually starts his technique by identifying objects and environments, before concentrating on activities and tasks. Ethnographic schools, on the other hand, always start by literally going into the situation, experiencing and subsequently analyzing it from within. In GTA there are various concepts needed to deal with the situation:

- Designated objects as analyzed by Walsh will be labeled *“things”* in our framework. These may be physical objects, or non-physical things like passwords, PINs, standard jokes, or stories. Things feature in tasks, sometimes being manipulated or created by tasks, sometimes as elements in conditions for starting or stopping a task. Things may act as symbols for roles (uniforms, portable phones, boy-scouts' handshake). Things may have attributes (opportunities for change) like a date of last inspection, a pointer to an owner.

- Things feature in a *structure*. In some cases it makes sense to represent things in a type hierarchy (e.g., in offices, there may be a type structure of paper fill-in forms). Often things provide places for other things (a letter may contain a header, a content, several appendices, and a signature)
- *Events* model things that happen in the task world over which the actor does not always have direct control, e.g. lightning strikes, a power failure, a colleague has become sick, a burglary, arrival of new mail etc. When analyzing highly complex situations such as a cockpit or a construction yard the event may prove very useful in modeling the rapidly changing situations.
- All tasks in a work domain are performed in an *environment*. The environment can be conceived as a thing, which has static and dynamic aspects. Environments are characterized by the things, the agents and roles that live in it but also more dynamically by the events that have taken place and those that may occur in future.

The three viewpoints described in this section are described in shared concepts that are interrelated. In the first versions of GTA, we focused on representing the separate viewpoints. Work can be represented in task structures that mainly illustrate the hierarchic nature of the relation between higher order tasks and subtasks. In the same way the organization of agents or situations can be represented in role type/containment hierarchies. From the start GTA provided object oriented templates for representing the single concepts in their relation to the other relevant concepts, much like MAD for tasks (Scapin and Pierret-Golbreich, 1989) and ETAG for things (Tauber, 1990). In these templates, a concept like a thing is related to other things (sub type relation, contains relation), to actors and roles (ownership), to tasks, etc.

NEED FOR REPRESENTATION OF PROCESS

Task analysis techniques often focus on a single viewpoint (HTA, MAD, GOMS) or at least restrict themselves to separate representations for different viewpoints: ETAG uses separate structure descriptions for things (the UVM, or 'Users' Virtual Machine') and for tasks (the dictionary of basic tasks), and so did GTA in its ancient format. Experience with GTA in a variety of actual design situations (office work, construction hall, mobile communication, business processes industrial process control) showed that GTA in its original versions lacks possibilities that are essential to completely model relevant task knowledge. Like other modeling techniques, GTA did not allow adequate handling and representation of time aspects and synchronization, parallelism and dataflow. GTA has been used in practical situations, though. Van der Veer, Hoeve, and Lenting (1996) applied GTA to a task analysis of a large public administration system, and tried to solve these problems by elaborating the concept of constructors in hierarchical representations. Traunmueller and Wimmer (1997) applied GTA for the same type of complex systems and pointed to the need for workflow type analysis concepts. Other real life applications of GTA are currently in progress, including the analysis and modeling of the Italian Railway traffic control system (Wimmer, in preparation) and the re-design of a commercial security system for banks, power plants, and airfields (van Loo, in preparation). In all cases the method proves to be very useful for collecting task knowledge, for modeling and for analysis as well as for envisioning the future task world. The various types of representation that are part of the GTA approach (see below for some examples) are suited for use in multidisciplinary design teams. In all cases, however, there was a need for expanding the graphical and formal representations in order to adequately provide insight in the temporal aspects of task decomposition as well as in the relation between various tasks that may be in process in parallel.

A representation technique that seems promising in this respect (as pointed out by Traunmueller and Wimmer, 1997) may be found in workflow management tools. Here, tasks are in fact represented by streams of entities that travel between situations or roles. Workflow representations allow the distinction between three types of organizational processes (Grudin, 1994): *material* (to be interpreted as products of the process), *information* (where information is represented in its function of driving the material process), and *human* (the coordination of people and organizations in respect to the steering of information and material processes).

GTA EXTENDED

We try to overcome the problems by a two-fold extension to the original GTA method: addition of some concepts to the conceptual framework, and inclusion of new representation formats that allow modeling complex processes. Our solution exists of including certain concepts from workflow modeling in the conceptual framework of GTA:

- *Time*, related to tasks in the form of deadlines, interval and frequency, and process time. An absolute time value can enforce the start of a certain task execution, a relative time can trigger execution of a task (start within 1 hour from a certain event of from the execution of another task). A task can be stopped at a certain absolute time, or after a certain interval. In this way time may feature in both starting and stopping conditions for tasks.

- *Input and output containers*, relating data and object transfer between tasks. Tasks often concern the handling of things. Things may be created or removed, things may be changed by changing the value of attributes, by changing the set of (other) things they contain, or by changing their location (inside other things). In all these activities, other things and "data" play a role, both as input for the task, and as results that will be transferred to subsequent tasks, whether these will be initiated by the same agent or by others. To this end, the representation of tasks needs to be extended to contain the relevant things and data.
- *Workflow events*, which are events resulting from a task within the system that controls the execution of another task. Tasks not only are related by way of decomposition into (sets of) subtasks. Often tasks result in other tasks being enforced or enabled to start or to end, even if other actors are concerned (sending an email to another person). An event can occur due to the completion of a task or of a certain action within a task, as the result of a certain condition during the performance of a task or a condition that emerges in the task environment or even because something originates from the outside world, like a telephone call or an earth quake.

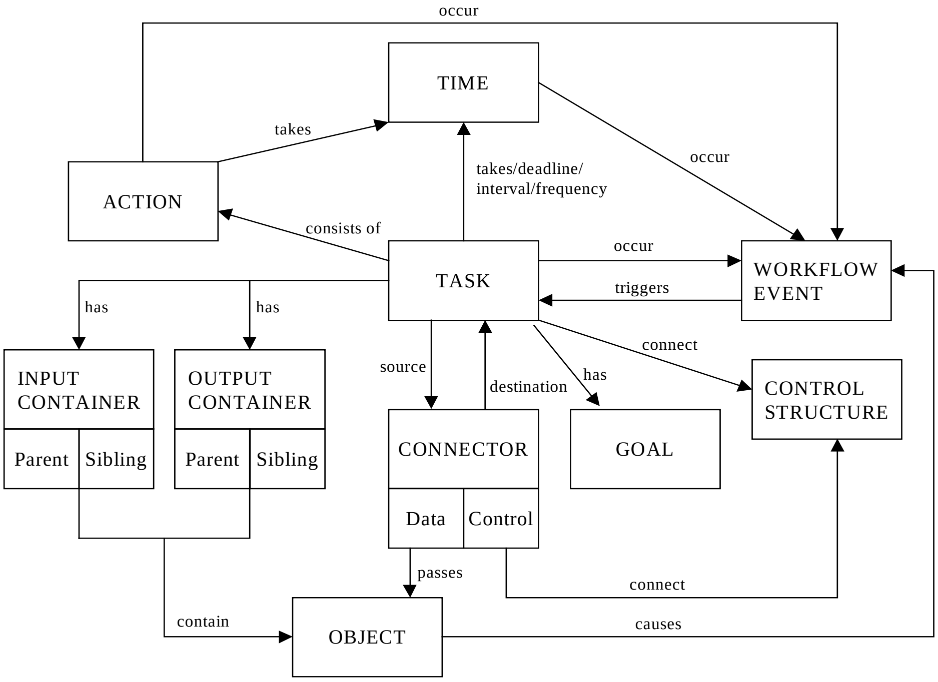


Figure 2 Extended conceptual framework of GTA

In Figure 2 we present the structure of the task view in the new conceptual framework of GTA. Concepts are represented by boxes and labeled arrows indicate directed relations. The ***task*** concept is still central, and each task has a single ***goal*** (a goal may be reached through performing various alternative tasks, though) and each task may be performed or delegated by performing ***actions,*** as in the previous version of GTA. The new entities in this view are:

- ***Time***, attributed to task through various relations:
    - a task may be characterized by the *time* it takes;
    - a task may have a *deadline*;
    - a task can normally be performed with a certain *frequency* (per time interval), or, alternatively, a task my be specified to occur at a certain regular *interval.*
- ***Input containers*** and ***output containers***, that in itself may be structured into *parents* and *siblings*, and that may contain ***objects*** (also referred to as "things". This entity did not change from the previous definition). A parent container contains objects which have been received from higher order tasks, or which will be send to higher order tasks in the task hierarchy, the sibling container contains objects that are transferred between tasks of the same level. There is no need to represent transport of data or objects to subtasks, for at the level of describing a certain task one should not bother about its subtasks.
- ***Connectors***, which are of two distinct types:

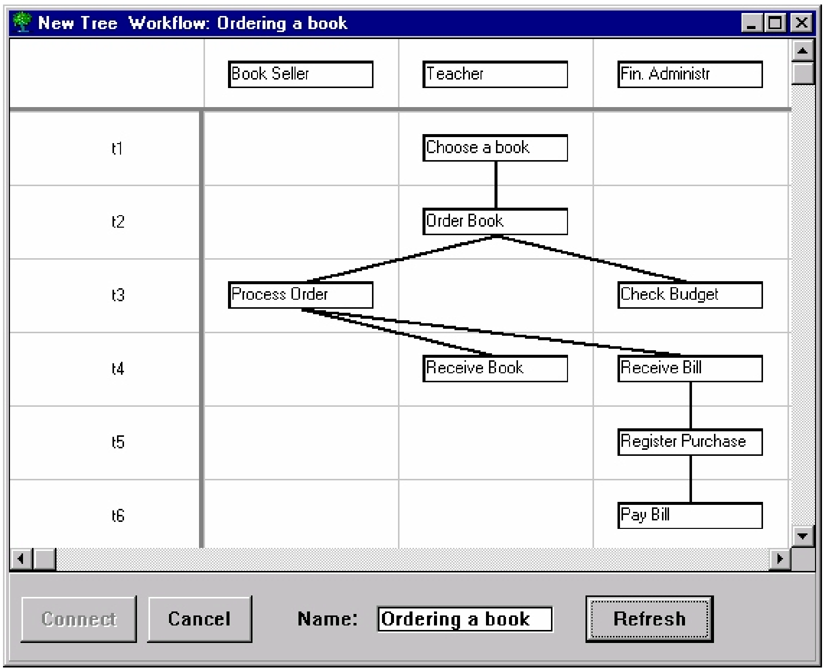
- ***control connectors*** allow the representation of forced (i.e., triggering) or enabling precedence between tasks. Each connector refers to a *control structure* as used in workflow. Between two tasks there can be a *delay*. This delay is an attribute of the control connector which connects the two tasks with each other;
- ***data connectors*** represent the passing of objects or date between data containers of different tasks.

A task can be the *source* or the *destination* of both types of connectors, allowing us to represent both the control flow and the data flow between tasks.

- ***Workflow events***, representing *triggering* or *enabling* relations between tasks and between data or objects and tasks. A task may cause an event to *occur*, and so may do the condition of an object (e.g., an email is received) or a time condition.

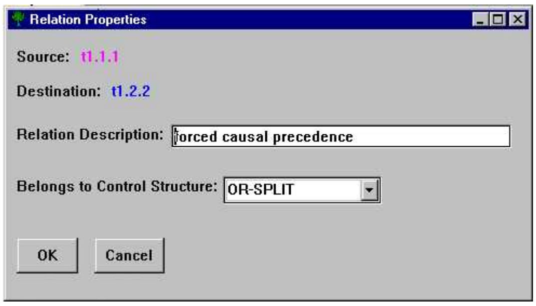
The extended framework allows new characteristics for the technique of modeling complex processes:

1. each process can contain several different processes (in parallel, synchronous, alternative, etc.);
2. each process can be represented as a relation between time and either roles (or actors) or objects (including data, situations, or locations);
3. consequently, each process can be represented as a directed graph of tasks, each of which related to roles or objects;
4. a task or subprocess can be part of several processes.



Name: Ordering a book

Figure 3 Experimental Workflow viewer



Source: t1.1.1

Destination: t1.2.2

Relation Description: forced causal precedence

Belongs to Control Structure: OR-SPLIT

Figure 4 Setting relation properties

Figures 3 and 4 give some examples of what we can model now with the extended framework. For instance, at time t3 there are parallel processes "Process Order" and "Check Budget". Each process is represented in figure 3 related to both time t_i and roles (bookseller, teacher, financial administrator). Figure 3 is a directed graph: connecting lines specify time-sequence of subtasks, as far as tasks are performed conditionally on the previous tasks(s). Detailed information of the relation between related tasks can further be represented as in figure 4 where the control structure and the type of relation are specified.

TOOL SUPPORT FOR PROCESS MODELING

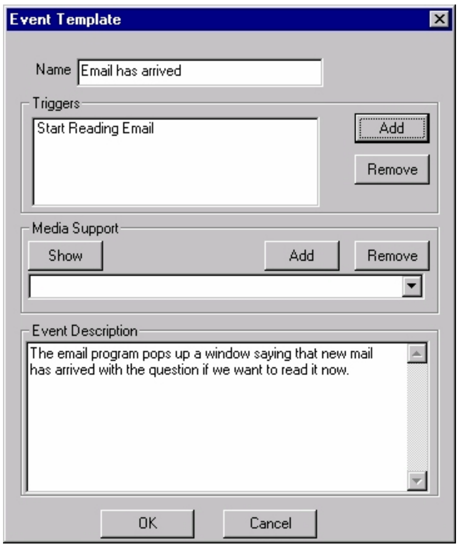
The extended framework claims to be suitable for specifying process. We have therefore built an experimental workflow viewer that is part of our tool EUTERPE (van Welie, van der Veer, Eliens, 1998). It is a graphical task analysis tool that can be used to enter task analysis data and to analyze them. It is based on the concepts and relationships of GTA which are internally represented using a logical programming language. The logical representation is on an abstract level and it does not imply any graphical representation. However it is rich enough to accommodate the extraction of the necessary information to generate commonly used representations such as tree structures, process flow graphs or templates. Figure

3 shows an experimental version of a workflow viewer based on the new concepts and relations of the described extensions of GTA.

The workflow viewer shows each process in a separate window that also offers graph-editing facilities. A process can be chosen from a list of process or by indicating the starting task of the process. When a task is the starting task of more than one process the user is presented with a list of processes from which he can select the relevant process. Tasks or objects can be shown in their relationships with roles or agents. Figure 3 shows some tasks of a book ordering system and the involved roles.

EUTERPE offers several representations that are all generated from the same data which guaranties consistency among the different representations. Representations include task trees, object hierarchies, templates with detailed information about entities such as tasks and objects, entity list and process graphs. In order to incorporate ethnographic data it is also possible to attach video fragments, images or sounds to any of the entities.

Figure 5 shows an event template that allows the event properties to be specified and figure 6 shows a task template that allows some basic time aspects to be specified, either in the conditions or in the frequency and duration properties. For other concepts similar templates are available. The templates are designed to show the concept properties but also the relationships with other concepts. For instance, the task template shows the used object and related events that trigger this task. Our tools are available for downloading at http://www.cs.vu.nl/~martijn/gta/.



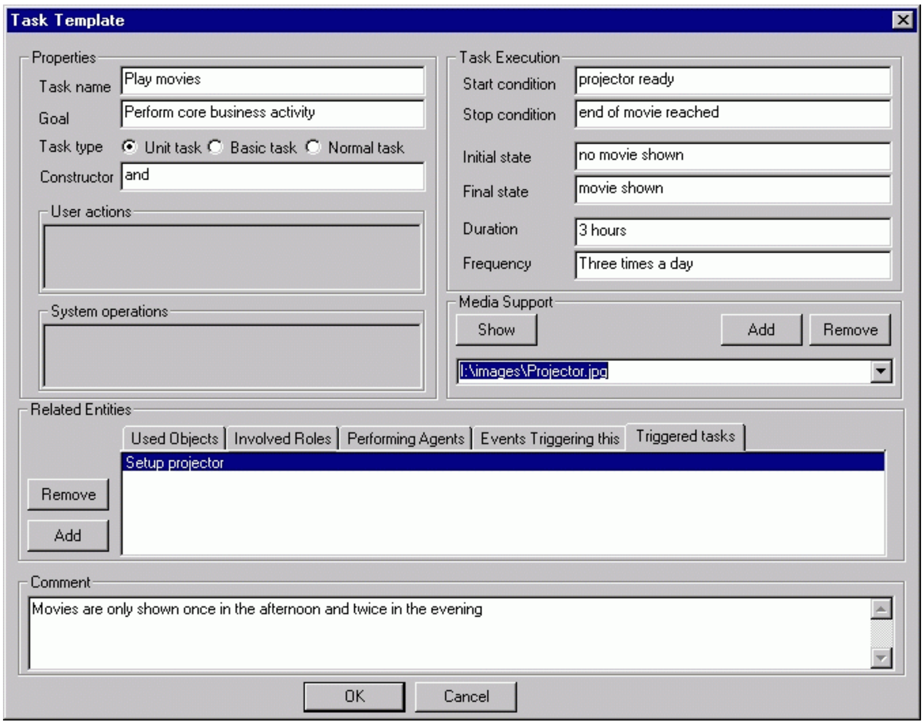
Name: Email has arrived

Triggers: Start Reading Email

Event Description: The email program pops up a window saying that new mail has arrived with the question if we want to read it now.

Figure 5 Specifying an event

CONCLUSIONS



Task name: Play movies

Goal: Perform core business activity

Task type: ● Unit task ○ Basic task ○ Normal task

Constructor: and

Start condition: projector ready

Stop condition: end of movie reached

Initial state: no movie shown

Final state: movie shown

Duration: 3 hours

Frequency: Three times a day

Media Support: I:\images\Projector.jpg

Triggered tasks: Setup projector

Comment: Movies are only shown once in the afternoon and twice in the evening

Figure 6 Specifying a task

In this paper we described how we extended our conceptual framework GTA with support for modeling complex processes. By using workflow representations we add new and useful information to the classic hierarchical representations used in task analysis. Apart from a workflow view we motivated the use of other views that may prove useful in representing tasks models. Our design environment EUTERPE is developed to support these views.

REFERENCES

Card S.T., Moran T.P., and Newell A. (1983) Psychology of human-computer interaction. Erlbaum, Hillsdale, NJ

D. Diaper (ed) (1989) Task analysis for human-computer interaction. Ellis Horwood, Chichester

Grudin J. (1994) Groupware and social dynamics. Communications of the ACM, 7 (1), 92-105

Hirschheim R., Klein H.K. (1989) Four paradigms of information systems development. In: Communications of the ACM 32 (10), 1199-1216

Johnson P. (1989) Supporting system design by analyzing current task knowledge. In: D. Diaper (ed) Task analysis for human-computer interaction, Ellis Horwood, Chichester

Johnson P. (1992) Human-computer interaction. McGraw-Hill, London

Johnson P & Johnson H & Waddington R & Shouls A (1988) Task-Related Knowledge Structures: Analysis, Modeling and Application. In: D.M. Jones and R. Winder (eds) People and Computers IV, Cambridge University Press, 35-62

Jordan B. (1996) Ethnographic workplace studies and CSCW. In: D. Shapiro, M.J. Tauber, R.Traunmueller (eds) The design of computer supported cooperative work and groupware systems. North-Holland, Amsterdam. 17-42

Kirwan B. and Aisworth L.K. (1992) A guide to task analysis. Taylor and Francis, London

van Loo R.J. (in preparation) SeCrecy, Sensitivity, Security: TA of an existing security system called CSS. In: G.C. van der Veer (ed) Task Analysis - state of the art in European approaches. Springer, Heidelberg

Nardi B. (ed) (1995) Context and Consciousness: Activity Theory and Human Computer Interaction. MIT Press, Cambridge MA

Rizzo A., Wimmer M. (in preparation) Holistic design in railway traffic control: experiences from work analysis and mock-up design. In: G.C. van der Veer (ed) Task Analysis - state of the art in European approaches. Springer, Heidelberg

Scapin D. and Pierret-Golbreich C. (1989) Towards a method for task description: MAD. In: L. Berlinguet and D. Berthelette (eds) Work with display units 89. Elsevier , Amsterdam

Sebillotte S. (1988) Hierarchical planning as a method for task-analysis: the example of office task analysis. Behaviour and Information Technology 7 (3), 275-293

Shapiro D., Tauber M.J., Traunmueller R. (eds) (1996) The design of computer supported cooperative work and groupware systems. North-Holland, Amsterdam

Tauber M.J. (1990) ETAG: Extended task action grammar. In: D. Diaper (ed) Human-computer interaction - Interact'90. Elsevier, Amsterdam, 163-168

Traunmüller R., Wimmer M. (1997) Extending the Van der Veer Approach to Cooperative Work: A bridge between Groupware Task Analysis and Workflow Design . Proceedings of DEXA September1997, Toulouse France, 431-439

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., van Vliet J.C., and Lenting B.F. (1995) Designing complex systems - a structured activity. In: G.M. Olson, S. Schuon (eds) DIS '95 Symposium on Designing Interactive Systems. ACM Press, New York, N,. 207-217

van der Veer G.C., Lenting B.F., Bergevoet B.A.J. (1996a) GTA: Groupware Task Analysis - Modeling Complexity. Acta Psychologica (91), 297-322

van der Veer G.C., Hoeve M., and Lenting B.F. (1996b) modeling complex worksystems - method meets reality. In: T.R.G. Green, J.J. Canas, and C.P Warren (eds) Cognition and the worksystem- 8th European Conference on Cognitive Ergonomics. EACE, INRIA, Le Chesnay cedex. 115-120

van Welie M., van der Veer G.C., Eliens A. (1998) EUTERPE - Tool support for analyzing cooperative environments. In: T.R.G. Green, L. Bannon, C.P. Warren, J. Buckkley (eds) Cognition and Co-operation - 9th European Conference on Cognitive Ergonomics. EACE, INRIA, Le Chesnay Cedex, 25-30

Walsh P. (1989) Analysis for task object modelling (ATOM). In: D. Diaper (ed) Task analysis for human-computer interaction. Ellis Horwood, Chichester,. 186-208

Wood L.E. (ed) 1998, User interface design. CRC Press, Boca Raton, Florida.