STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **XLIX**, Number 2, 2004

TASK MODELING IN SYSTEMS DESIGN

ADRIANA-MIHAELA TARŢA

Abstract. This paper aims to provide a discussion of how model-based approaches and the related tools can be used to obtain usable systems. More than that, in the last years the design of software systems is regarded as a multisciplinary area, where knowledge from psychology, sociology, etnography are interrelated. This paper will show how some techinques specific to these disciplines can be used in user interfaces design.

1. Introduction

People are often using models in their everyday life. Everytime when a more complex problem is encountered, the way of dealing with the complexity is by using models, i.e. identifying the main aspects that should be taken into account and their relations. It is not suprising that in the software design domain, this approach has been adopted and now is a technique often used to represent the various aspects of the problem that must be solved (domain, data, interaction, etc.). The models' goal is to provide a structural description of the relevant information. The purpose of model-based design is to identify high-level models which allow designers to specify and analyse interactive software applications from a more semantic-oriented level rather than starting immediately to address the implementation level. This allows them to concentrate on important aspects without being confused by the implementation details. Models are capturing the semantic meaningful aspects and allow the designers to manage more easily the increasing complexity of interactive systems [Pat04]. The most succesful modeling technique used in software engineering is UML (Unified Modelling Language), which focus on modeling the objects composing a system and then modeling of activities that

Received by the editors: December 10, 2004.

2000 *Mathematics Subject Classification.* 68P99.

1998 *CR Categories and Descriptors.* **D.2.2** [Software]: Software Engineering – *Design Tools and Techniques*; **H.5.2** [Information Systems]: Information Interfaces and Presentation – *User Interfaces* .

manipulates the system. The object-oriented approaches are successful at engineering the software implementation level. One of the basic usability principles says:"focus on the user and their tasks", that's why a new trend have gained attention in the last years, when the comfort of people, the ergonomic aspects of people work are emphasized more and more. This new trend is called task modeling. The task oriented approaches focus also on objects and activities, but the difference from object-oriented approaches is that the task-oriented approaches first identify the tasks that users perform and then the object manipulated.

2. Task Analysis and Modeling

Before discussing the task analysis and modeling techniques, we have to understand the meaning of "task" concept. A task is an activity that should be performed in order to reach a goal. A goal is a desired modification of state or an inquiry to obtain information on the current state of an object (system)[GM02].

Task models' goal is to identify useful abstractions highlighting the main aspects that should be considered when designing interactive systems. The main advantage of task models is that they represent the logical activities that an application must support. In order to better understand the user, his work and his expectations, there are two types of task models: descriptive and prescriptive. The descriptive task models describe the way in which the task is performed currently (even in an manual way, or supported by another system). The prescriptive task model describes how the task should be supported by a new developed system [Pri03].

The descriptive task models are developed by psychologists, etnographers and domain experts, using data aquisition techniques like: interviews, video recording or direct observation. After data acquisition the information are gathered, analyzed, and a descriptive task model is generated.

The prescriptive task model is build on the descriptive task model, but it is restructured, in order to include the modification introduced by the use of new technology (some redundant tasks will be eliminated, some user tasks will change in application tasks or interaction tasks, when the user and the system perform a task in a collaborative manner). The prescriptive task model is conceived by the designer teams and human factor experts, in order to assure a high level of usability.

A task model in the design of an interactive system describes a set of activities that users intend to perform while interacting with the system. There are two types of task models: *the system task model* that provide information about how the designed system requires tasks to be performed, and *user task model* which is

how users expect to perform their activities. It is desirable that these two models be very similar, otherwise some usability problems will be present.

Task models are essential in the design of interactive systems because they represent the logical activities that should support users in reaching their goals. From informal representations only (like mock-ups or scenarios) the designers won't have the necessary information to support the design decisions. Task models represent the intersection between user interface design and more systematic approaches by providing designers with means of representing and manipulating an abstraction of activities that should be performed to reach user's goals.

Task models are built after task analysis is performed. Task analysis aims to identify the relevant tasks and how activities are performed currently. Task models describe the semantic and temporal relations between the identified tasks.

3. Task Modeling Techinques and Tools

Task analysis and modeling is an old method of describing and analyzing the structure of work and was first used in psychology. The first important method of task analysis was HTA (Hierarchical Task Analysis), developed in 1960 [DFAB93]. The method represents the structure of task in a hierachical decomposition. The representation of task structure had two forms: a textual description, where indentation and numbering of task/subtasks were used, and a graphical representation, using trees. The order of task performance was given by plans which describe the performance order using the tasks/subtasks numbers.

This approach have been used lately in other task analysis techniques like GOMS (Goals, Operators, Methods, and Selection Rules), where the tasks and it's goals were represented hierarchically, using a textual description.

Even in the last years, the hierarchical decomposition of tasks is considered the best suited method to describe task models. Because of the interdisciplinary effort for the design of interactive systems, the graphical representation was adopted. Task analysis methods like GTA (Groupware Task Analysis)[vdVvW00] and CTT (ConcurTaskTrees)[GM02] use a tree representation of task in their associated tools. GTA is a method that starts from the basic idea that nowadays people are performing their work in a collaborative way, so the work analysis is made for groups of people. People who are performing a common set of tasks form *roles*, and the system or the people is considered to be an *agent*. In task performance, agents manipulate *objects*. The performance of a task is triggered by *events* or by another task. In GTA the temporal relation between tasks can't be precisely specified. The method doesn't provide any formal mean of specifying such relations, providing

support only for the description of sequential tasks. These are the main concepts used to describe the task world and the temporal relations between tasks [vW01].

While the number of task analysis method is signifiant, there are only few tools that support the use of these methods and that generates task models. Only two tools are freely available on the Internet: EUTERPE, the task analysis and modeling tool that supports the use of GTA technique [vW01], and CTTE (ConcurTaskTrees Environment)[GM02] corresponding to CTT (ConcurTaskTrees) analysis and modeling method. EUTERPE provides functionality for describing the task models in a graphical manner and to store all the objects, responsible roles, triggered tasks and triggering events for each task. The models can be verified using a validation tool which verifies a set of constraints that must hold for every correct model (e.g. tasks that are never triggered, roles which aren't responsible for any task). EUTERPE also generates the work flow based on the triggering events or tasks, but the temporal description is quite ambiguous, because of the lack of formal methods for representing the time relations between tasks.

The problem of unambiguous specification of time relations is well solved in CTT (ConcurTaskTrees) that has an associated tool called CTTE (ConcurTaskTrees Environment). CTT use also a hierarchical representation of tasks using task trees, but the relations between tasks is described using LOTOS operators: choice, order independency, concurrent, concurrent with info exchange, CTT disabling, suspend/resume, enabling, enabling with info exchange [GM02]. In CTT descriptions are also considered the agents, roles and objects related to tasks. CTTE provides means of describing single user task models and also cooperative task models. The tool provides also a simulator that helps designers understanding the logical order of tasks' performance, because when having task models for complex systems is difficult to verify all the possible paths in task execution.

From our experience in using the above mentioned tools the conclusion is that each of them has its advantages and disadvantages. EUTERPE supports the phase of task analysis because of the various information about task's world which can be stored by filling the task templates forms and associating documentation fragments (video fragments, audio fragments, sketches). From a GTA approach, we can say that EUTERPE supports very well the building and the validation of task model 1 (the descriptive model of the task). On the other hand, CTTE supports the building, verification and simulation of task model 2 (the prescriptive one). The use of temporal operators is an essential feature and opens a road to the specification of presentations and interactions of the modeled system.

4. Task Modeling in the design of interactive systems

In the design of interactive systems the task analysis can be used with different goals:

- requirement analysis - when through a task analysis designers identify requirements that should be satisfied in order to obtain an useful system;
- design of interactive applications - the information from task models is used to better identify the interaction techniques and the presentations of the application (in this case the modeling technique should provide temporal information about the logical order of tasks);
- usability evaluation - the system task model and the user task model are compared in order to get information about the matching between these models; also, having a structured task model some techniques like KLM (Keystroke Level Model) can be applied to get information about the time needed to perform a task - this kind of approximation may be used also to compare different task models addressing the same problem.

Task models are built for many different situations: usually, a task model is built when designing a new application with the goal of obtaining precise information about: the order of task performance, objects from the domain manipulated in the task performance process, agents and roles responsible for the task performance, events triggering task performance, preconditions and postconditions for task performance. Also, a task model can be built for an existing application, in this case the goal is to understand the underlying design, analyse its limitations, and solutions to overcome them. Task model can address the problem of designing an entire application, or just a part of it.

4.1. **User Interface Design Based on Task Analysis.** Having stored in a repository all the information acquired in the task analysis phase, having the structured representation of activities implied by a task performance in a task model and having the relations between activities, objects manipulated, agents or roles responsible for each task, using some additional information about the type of tasks (editing, selection of one/multiple choices, responding to alerts, etc.) and some descriptions about the objects manipulated (numerical values, texts, etc), we have the possibility of automatic determination of the main presentations of the interactive system (the kind of widgets needed in a presentation); also, from the task model we know the navigational path in the interactive system. The information obtained might support the process of building an abstract specification of the user interface (using specification languages like XML, XIML (eXTended Interface Markup Language) or UIML (User Interface Markup Language)) which

can be then rendered in a concrete user interface. Task models play a key role when the goal is to develop multiple user interfaces for different devices. In this situation, the task models capture all the relevant requirements at the task level and using some tools the abstract specifications, and then the concrete specification for different platforms can be generated. The structure of task model provides guidance for the navigational structure of the concrete user interface and for the grouping of the object manipulated in the task performance [LM03].

5. Problems with Task Modeling Techniques and Tools

Although the use of task modeling approaches even in real case study have proved its efficiency, there are some aspects criticised in the domain literature [vW01, Pat04].

One problem related to the development of task models in software design process is the extra time needed. Indeed, the process of analyzing people work and of representing the most important aspects in a structured model need a time for data acquisition and analysis, but now there are tools that help designers in building such task models or functionality of modeling tools (e.g. EUTERPE can display multimedia files registered in the working environment that helps designers in building task models, the CCT task models can be built from an informal specification (scenarios) using the El-TaskModels tool which helps designers identifying roles, objects, tasks from informal descriptions [GM02]. Also, if the goal is to develop a complex application, it is hard to manage complexity without creating models that provides the semantic and temporal information. The graphical representation of models takes advantage of the readability for different kind of specialists involved in the design process.

Another criticism of task modeling technique is that they aren't applicable for creative tasks, because in that special case there is no structured task model. Task models support only functionality where users want to achieve a goal and the activities needed can be modeled in a structured fashion.

Task models are often used for the design of user interfaces and interaction styles. Task models lead to verb-noun interaction style which is not preffered by some users who like first select an object and then the actions on that objects. We have to mention that these are two different ways of achieving goals, and task models can be used for implementing both interaction styles, even the verb-noun style is more intuitively extracted from models.

6. Conclusions

We have presented in this article a different approach that can be used in the design of usable software systems. This approach focuses on user and user's tasks and from the analysis of this information a task model is built that supports the further design of interaction and presentations of the designed system. We have presented also some available tools that support the task analysis and design. Because this approach is still at the beginning the number of such tools is reduced, but from the experience of using them we know now what are the requirements for such a tool: the presence of a graphical editor for task trees, support for multiple representations (formal and informal), consistency between representation, support for handling different media types in the documentation process of building models (audio files, video fragments), support for models' verification and support for simulation. Most of these requirements are met by the above mentioned presented tools, but there is an essential aspect that isn't covered by any of them. While building a task model, there are several persons implied in this activity, each of them bringing it's own contribution to the task model. There isn't any tool that offers support for collaborative building of tasks that implies the management of versions and changes. We consider that this is an important aspect for future research and development of task analysis and modeling tools. Using the above mentioned tools we have found the difficulties in using the tak analysis methods and the corresponding task analysis tools. Our future work will focus on developing a new task analysis and modeling tool based on the GTA approach. Some improvements will be made on the specification of temporal relations between tasks following the ConcurTaskTrees approach. The tool should have the functionality of automatic generation of an abstract user interface specification from the task models.

References

[DFAB93] Alan Dix, Janet Finlay, Gregory Abowd, and Russell Beale. *Human-Computer Interaction*. Prentice Hall, 1993.

[GM02] Carmen Santoro Giulio Mori, Fabio Paternò. Ctte: Support for developing and analyzing task models for interactive system design. *IEEE Transactions on Software Engineering*, Vol. 28(9), Sepember 2002.

[LM03] C. Santoro L. Marucci, F. Paternò. *Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces*, chapter Supporting Interactions with Multiple Platforms Through User and Task Models, pages 217–238. Wiley and Sons, 2003.

[Pat04] Fabio Paternò. Model-based tools for pervasive usability. Technical report, University of Pisa (Italy), 2004.

[Pri03] Costin Pribeanu. *Introducere în interacţiunea om-calculator*, volume 1 of *Interacţiunea om-calculator*. Matrix Rom, 2003.

[vdVvW00] Gerrit van der Veer and Martijn van Welie. Task based groupware design: putting theory into practice. In *DIS '00: Proceedings of the conference on Designing interactive systems*, pages 326–337. ACM Press, 2000.

[vW01] Martijn van Welie. *Task-based User Interface Design*. PhD thesis, Vrije Universiteit, 2001.

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Department of Computer Science, 1, M. Kogalniceanu Street, RO-400084 Cluj-Napoca, Romania, EUROPE

E-mail address: adriana@cs.ubbcluj.ro