

Developing task models from informal scenarios

Fabio Paternò

CNUCE-C.N.R.

Via S.Maria 36

56126 Pisa, Italy

+39 050 904052

f.paterno@cnuce.cnr.it

Cristiano Mancini

CNUCE-C.N.R.

Via S.Maria 36

56126 Pisa, Italy

+39 050 904052

c.mancini@cnuce.cnr.it

ABSTRACT

The difficulties that designers and developers often have during the development of task models is a strong limitation to their use. Usually the main problem is to identify what is useful for the development of such task models from a lot of informal information. We propose a method, with a related tool, supporting the development of task models able to describe concurrent dynamic activities using an informal scenario as input.

Keywords

Task models, Automatic support for designers, Formal and informal representations.

THE PROBLEM

In the last years a strong interest in using task models has grown because they have been found useful not only to improve an understanding of the application domain considered but also to support an effective design and to perform usability evaluation. Various experiences have shown that automatic tool support, in the use of such models during the various phases of the design cycle, is helpful because it allows designers to easily identify possible solutions while still leaving them the possibility to choose the most suitable one and to tailor such solutions to the specific case study considered.

Some notations to specify task models have been proposed (examples are UAN [1], ConcurTaskTrees [2], the GOMS family [3]), they differ for the type of syntax (textual versus graphical), the level of formality, the richness of operators that they offer to designers. However they all share a common problem: many designers often found them difficult to apply, especially designers working in industries that often do not have strong background in structured methods and have hard time deadlines in their projects. We have been confirmed of such problems in the MEFISTO project that involves both research and industrial sites in the design and development of usable and safe user interfaces for Air Traffic Control

applications.

THE PROPOSED SOLUTION

When approaching the design of a new application or the re-design of an existing application, designers have often a lot of informal information available: documentation concerning existing applications, notes from meetings with users, requirements provided by customers, and so on. They have to refine this material to identify the task structure underlying the existing application to analyze or that corresponding to the new application to design.

Scenarios are a well known technique in the HCI field often used during the initial informal analysis phase. They provide informal descriptions of a specific use in a specific context of an application. A careful identification of a meaningful scenario allows designers to obtain a description of most of the activities that should be considered in a task model. The main difference between a task model and a scenario is that a scenario indicates only one specific sequence of occurrences of the possible activities while the task model should indicate all the possible activities and the related temporal relationships. The use of some tool support in the task elicitation phase was proposed in [4]. Here we propose a different solution oriented to support the development of task models expressed in ConcurTaskTrees, a formal notation with a wide variety of temporal operators supporting user interface design and usability evaluation that we are using to support the design of interactive safety-critical applications.

We start with an informal description of a scenario. The scenario should be selected so as to include performance of most of the main activities involved by the application considered. It can be either the description of a specific use of an existing system or an envisioned use of a new application to design depending on what the designer's goal is. Next the designer can load such a description in the environment provided by our tool and select the words related to activities (such as detection of conflict, paper strip's update, sends clearance) and add them to the list of tasks. The names of such tasks can be edited in order to make them more general. The designer can also interactively indicate how to allocate the performance of the task: to the user (if only internal cognitive actions are required), to the application, to a user interaction (if the

performance consists in user interactions with some device). This is specified by selecting the icon associated with the task allocation chosen. In the scenario's description it is also possible to select the objects and indicate to what tasks they are associated. One task can manipulate multiple objects during its performance and one object can be manipulated by multiple tasks. In this way designers have an environment allowing them to rapidly identify tasks, objects and their relationships.



Figure 1: The environment supporting task identification.

The next step is to identify the structure of the task model. We split this activity into two steps: identify the hierarchical structure among tasks and define their temporal relationships. The input for this phase is the list of tasks identified with the scenario support. This list is not definitive. It can be further modified, for example to add new tasks whose purpose is to logically group a set of identified tasks that are semantically connected and share some temporal relationship. In the example considered (Figure 2) we can group the tasks considered in the scenario in two main logical activities (handling change of air traffic sector and handling traffic in a sector).

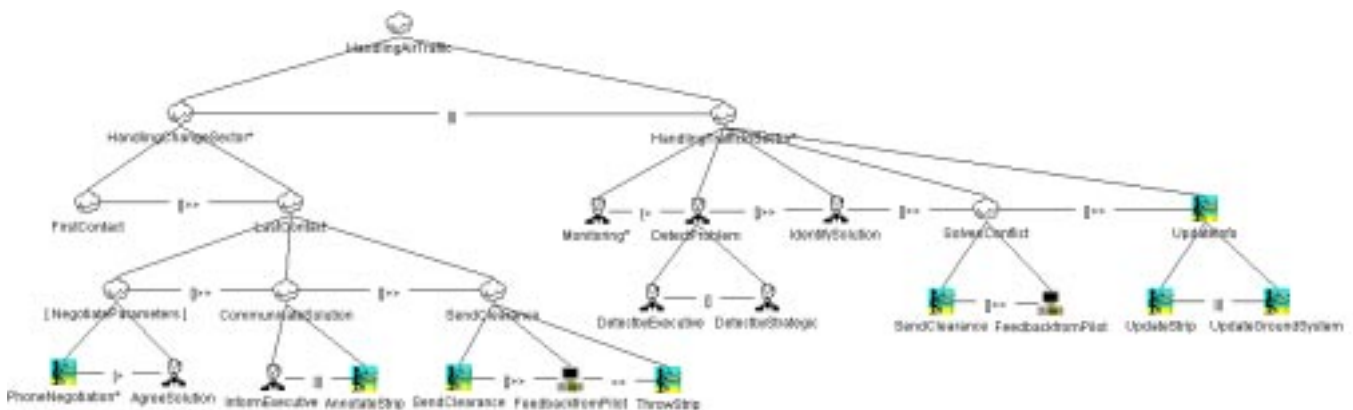


Figure 2: The task model corresponding to the scenario.

In our tool designers can activate an environment which has the list of tasks identified as input and allows designers to indicate a logical hierarchy among such tasks: from the list of identified tasks on the left side we can select a task and indicate its parent task on the right side.

We thus obtain a hierarchical task model that can be further edited by the existing ConcurTaskTrees editor (<http://girove.cnuce.cnr.it/ctte.html>). The difference is that now designers have not to start by scratch but they have available the hierarchy of tasks and most of the objects manipulated by such tasks have already been identified. Thus only the temporal relationships among the tasks have to be specified with the support of this editor. Figure 2 shows the task model obtained from the original scenario. At the first level there are two concurrent (\parallel operator) activities. They are decomposed into sequential (\gg or $[]\gg$ operator) or disabling ($[>$ operator) or mutually exclusive ($[]$ operator) activities.

CONCLUSIONS AND FUTURE WORKS

We have described a proposal for supporting the development of task models starting from informal descriptions of scenarios. The motivation was to give more support to designers in developing such models that are important in analyzing and evaluating interactive safety-critical applications. First experiences in our project showed the utility of this approach. Future work will be dedicated to further improve support for development of task models, for example helping designers to identify the temporal relationships among the tasks.

ACKNOWLEDGMENTS

MEFISTO is an Esprit Reactive Long Term Research Project (<http://girove.cnuce.cnr.it/mefisto.html>).

REFERENCES

1. R.Hartson, P.Gray, "Temporal Aspects of Tasks in the User Action Notation", Human Computer Interaction, Vol.7, pp.1-45, 1992.
2. F. Paterno', C.Mancini, S.Meniconi "ConcurTaskTree: a diagrammatic notation for specifying Task Models". Proceedings Interact'97, pp.362-369, Chapman&Hall.
3. B.John, D.Kieras, "The GOMS family of analysis techniques: Comparison and contrast". ACM Transactions on Computer-Human Interaction, Vol.3, N.4, pp.320-351, 1996.

4. R.Chung-Man Tam, D.Maulsby, A.Puerta, "U-TEL: A Tool for Eliciting User Task Models from Domain Expert", Proceedings IUI'98, pp.77-80.