CTTE: a tool for developing and analysing task models of cooperative applications

Fabio Paternò, Riccardo Lenzi, Ettore Torre, Livio Salconi

CNUCE-C.N.R., Via V. Alfieri, 1, 56010 Ghezzano, Pisa, Italy
Email: fabio.paterno@cnuce.cnr.it

Tool-support is strongly required in order to ease the use of task models and make them acceptable to a large number of designers. CTTE is a tool supporting task modelling and including a simulator for task models. This tool can be useful to better analyse the dynamic behaviour of task models including those for cooperative applications.

Limitations of current approaches

- Visual tools do not support effective design
- UML is oriented to the system design
- Tools for the development of multi-user applications are difficult to use

Usually task models are developed for interactive dynamic applications where the possible tasks to perform depend on the current state of the application. When analysing an existent application or designing a new one it can be rather difficult for the designer to understand the dynamic behaviour resulting from the temporal relationships specified in the task model. The reason is that, especially for real applications, the number of ways in which the application can evolve is high and it is difficult to mentally remember the various temporal constraints among tasks and their possible effects. To support this analysis of the dynamic behaviour of task models, interactive simulators can be helpful. The basic idea is that at any time they show the list of enabled tasks, according to the constraints specified in the task model. The designer can interactively select one of them and the simulator shows what the enabled tasks, after the performance of the selected task, are. This interaction can be iteratively performed a number of times. This is a support that only a few tools provide [4].

In this paper, we want to present a further contribution whose novelty lies in the possibility of simulating task models of cooperative applications when the resulting behaviour depends on the interactions of a number of users. To provide such support is important because the increasing spread and improvement of Internet connections makes possible the use of many types of cooperative applications. Consequently, tools supporting the design of applications where multiple users can interactively cooperate are more and more required.

SIMULATING TASK MODELS

The task models considered are structured in a hierarchical fashion according to the ConcurTaskTrees notation [5], with a rich set of operators to describe the temporal relationships among tasks. In addition, for each task further information can be provided such as its type, the category (indicating how the performance is allocated), the objects that it requires to be manipulated and attributes, such as frequency of performance. In this approach, when there are cooperative applications the task model is composed of various parts. There is one task model for each role involved and a cooperative part. The purpose of the cooperative part is to structure the cooperative tasks (those tasks that must be performed by two or more users). Thus, they are decomposed until we reach tasks performed by a single user. These single user tasks will also appear in the task model of the associated role. They are a sort of *connection* task between the single-user parts and the

cooperative part. The tool allows different ways to browse the task model. The designer can interactively select the pane associated with the part of the task model of interest (in Figure 1 there is an example with a task model composed of a cooperative part and two roles, Customer and sales Representative). In addition, when a connection task is selected in the task model of a role then it is possible to automatically visualise where it appears in the cooperative part and vice versa.

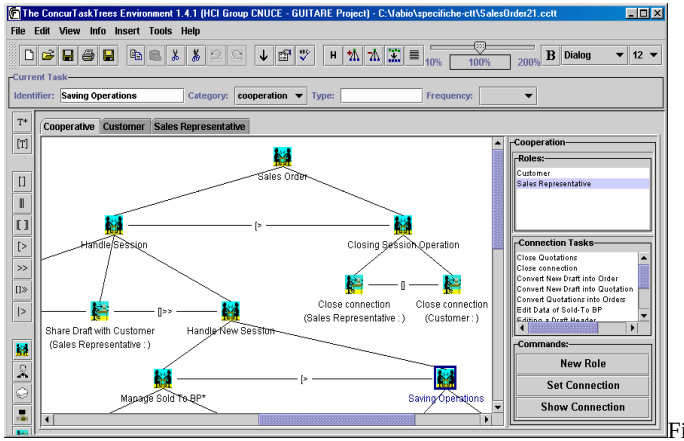

Figure 1: The editor of task models for cooperative applications.

Before starting the simulation, the tool automatically checks that the task model is complete and consistent. This means, for example, that the temporal relationships for all tasks are defined, that if a task is a leaf in the cooperative part then it should appear in a single user part and vice versa if a task is defined as a connection task in a single user part then it should appear in the cooperative part.

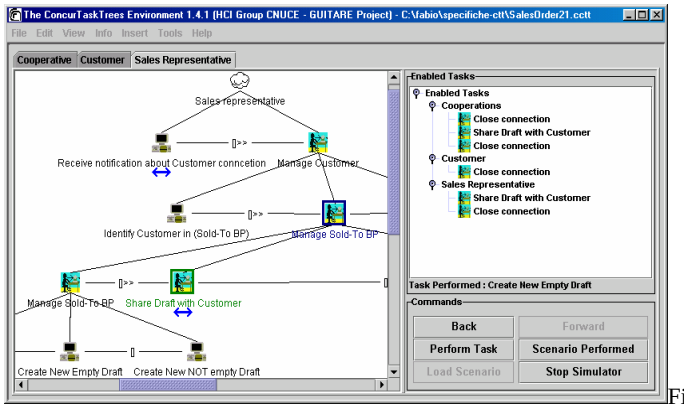

Figure 2: The interactive simulator of task models.

When the simulator is started, then the window on the right displays the list of tasks enabled (see Figure 2). The tasks that appear in such a list are basic tasks, tasks that are not further decomposed in the model. They are grouped according to the role to which they are assigned. In addition, the tasks that are part of cooperative tasks are listed again under the *Cooperations* label.

Then, the user can interactively select a task to perform and the simulator shows what the next enabled tasks are. The enabled tasks are also highlighted in the task model by a green frame. At any time, it is possible to go back through the performance of the tasks which means that the effect of the performance of the last task are undone and the list of enabled tasks becomes the same as that previous to the performance of the last task. At this point, the user can choose to go further backward in the task sequence or forward, either through the same path or a different one.

This interactive behaviour allows designers to dynamically identify abstract scenarios that can be useful in various phases. Such scenarios can also be saved in files and can be used to compare different task models. The tool is able to load a scenario from another task model and try to simulate the performance of the same sequence of tasks. If this is not possible, either because a task is not supported in the other model, or because the temporal relationships in that task model do not allow such a sequence, then it means that the scenario is not supported. This can be useful to compare the task model of an existing application with that of an envisioned application or two different task models that are related to two different designs.

CONCLUSIONS

The tool has been used in various projects for several application areas (air traffic control, enterprise resource planning, and museum applications). It has been found useful to clarify design issues, to support analysis and evaluation of design options. It can be freely downloaded at http://giove.cnuce.cnr.it/ctte.html.

ACKNOWLEDGMENTS

We gratefully acknowledge support from the GUITARE project (http://giove.cnuce.cnr.it/guitare.html).

REFERENCES

1. Baumeister L., John B., Byrne M., A Comparison of Tools for Building GOMS Models, *Proceedings CHI'2000*, ACM Press, 2000.
2. Bodart F., Hennerbert A., Leheureux J., Vanderdonckt J., A Model-based approach to Presentation: A Continuum from Task Analysis to Prototype, in *Proceedings DSV-IS'94*, Springer Verlag, pp.77-94.
3. Bomsdorf B., Szwillus G., Tool Support for Task-Based User Interface Design, *Proceedings CHI'99*, Extended Abstracts, pp.169-170.
4. Biere, M., Bomsdorf, B., Szwillus G., Specification and Simulation of Task Models with VTMB, *Proceedings CHI'99*, Extended Abstracts, pp.1-2.
5. Paternò, F., *Model-Based Design and Evaluation of Interactive Application.* Springer Verlag, ISBN 1-85233-155-0, 1999.
6. Puerta A., Cheng E., Ou T., Min J., MOBILE: User-Centred Interface Building, *Proceedings CHI'99*, pp. 426-433, ACM Press, 1999.
7. van Welie M., van der Veer G.C., Eliëns A., "An Ontology for Task World Models", *Proceedings DSV-IS'98*, pp.57-70, Springer Verlag, 1998.