CTTE: An Environment for Analysis and Development of Task Models of Cooperative Applications

Fabio Paternò, Giulio Mori, Riccardo Galiberti
CNUCE-C.N.R.
Via V.Alfieri 1
56010 Pisa, Italy
+39 050 3153066
{fabio.paterno, giulio.mori, riccardo.galiberti}@cnuce.cnr.it

ABSTRACT
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 an automatic environment that has been developed for this purpose. This tool can be useful to better develop and analyse task models and their dynamic behaviour including those for cooperative applications.

Keywords
Task models, Automatic Tools, Model Analysis and Development.

INTRODUCTION
The use of task modelling in supporting user interface design has long been the subject of research study. The main reason for such models is that after an informal task analysis where the main tasks, and their attributes, are identified, often there is a need to understand the relationships among the various tasks to better address the design and evaluation of interactive applications. However, one of the main problems in task modelling is that it is a time-consuming, sometimes discouraging, process. To overcome such a limitation, interest in the use of tool support [2] has increased in recent years. Various types of tools have been proposed. They can be classified into two main groups: those that aim to support editing and analysis of task models (such as GOMS tools [1]), and those that use task models to support design and generation of user interfaces (such as Mobi-D [5]). If we consider the former class of tools we can notice that they are mainly research prototypes aiming at supporting the editing of the task model and the analysis of the task relationships and related information. This is a useful contribution but further automatic support is required to make the use of task models acceptable to a large number of designers.

For example, 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 interactive dialogues can evolve following many paths and it is difficult to keep track of the various temporal constraints among tasks and their possible effects. Interactive simulators can be helpful to support the analysis of the dynamic behaviour of task models. The basic idea is that at any time the designer can interactively select one task and the simulator shows what the enabled tasks are after the performance of the selected task. This is a support that only a few tools provide [3]. In addition CTTE (ConcurTaskTrees Environment) gives the possibility of simulating task models of cooperative applications where the resulting behaviour depends on the interactions of various users. This support is important because the increasing spread of Internet connections makes possible many types of cooperative applications. Thus, tools supporting the design of such applications are more and more required.

ANALYSING TASK MODELS
The task models considered are hierarchically structured according to the ConcurTaskTrees notation [4], 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). In addition, when a connection task is selected in a role then it is possible to automatically visualise where it appears in the cooperative part and vice versa.

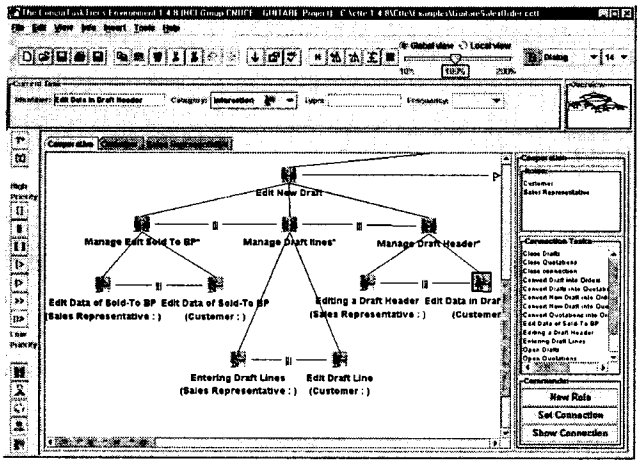

Figure 1: The editor of task models for cooperative applications.

SIMULATING TASK MODELS OF 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, 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.

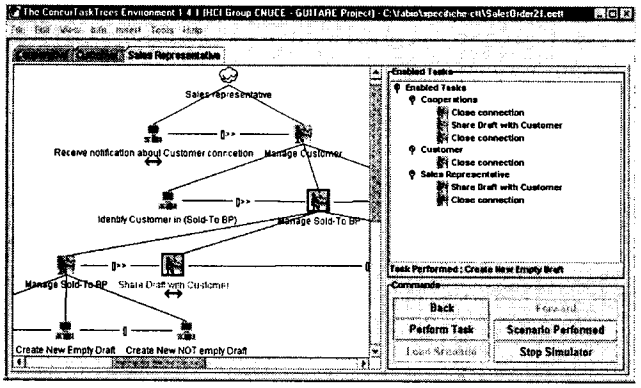

Figure 2: The interactive simulator of task models.

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 CTTE tool supports many features: to allow designers to edit task models and their layout; to use informal descriptions in modelling; to check completeness of the specification; to save specifications in various formats; to simulate task models; to compare task models and provide information on their structures; and to run scenarios.

It has been used in various projects for several application areas (air traffic control, enterprise resource planning, and museum applications). It helps designers clarify design issues and 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, pp.502-509, 2000.
2. Bomsdorf B., Szwillus G., Tool Support for Task-Based User Interface Design, *Proceedings CHI'99*, Extended Abstracts, pp.169-170.
3. Biere, M., Bomsdorf, B., Szwillus G., Specification and Simulation of Task Models with VTMB, *Proceedings CHI'99*, Extended Abstracts, pp.1-2.
4. Paternò, F., *Model-Based Design and Evaluation of Interactive Application*. Springer Verlag, ISBN 1-85233-155-0, 1999.
5. Puerta A., A Model-Based Interface Development Environment, *IEEE Software*, Vol.14, N.4, pp.41-47, 1997.