A tool to model user interaction in open problem solving environments.

N.K. Tselios, N.M. Avouris, M. Kordaki

University of Patras
ECE Department – HCI Group
GR-26500 Rio Patras, Greece
{ nitse, N.Avouris} @ee.upatras.gr, Kordaki@cti.gr

SUMMARY

A model-based approach in design and evaluation of computer-based open problem solving environments is presented in this paper. The functionality of a tool (CMT) that has been developed to support this process is also discussed. A number of models are defined according to the proposed approach: (a) a designer model and (b) user models as deduced from observation of user behaviour during field studies. Identification of discrepancies of the two models can lead to improvements in the design of developed prototypes. Examples of application of this design approach and the CMT tool are included in the paper.

KEYWORDS : Student task model, User interface design, Usability evaluation, GOMS, Open problem solving educational environments.

INTRODUCTION

One new challenging area of applications for the human-computer interaction community is that relating to the design of *open problem-solving educational environments*. These are computer-based learning environments that let students actively explore certain concepts while they are engaged in problem solving. These environments put emphasis in the active, subjective and constructive character of learning [11, 5]. In these environments the student's activity can not be reduced in a sequence of pre-defined tasks as in many other application areas and therefore evaluation of the correct answers is not straightforward, as in traditional learning environments.

There are many reasons for which use of model-based interaction design and evaluation approaches are suitable in this context [5]. However application of techniques such as GOMS and Task Analysis, requires adaptation of existing Human Computer Interaction research findings, which primarily concern users engaged in typical workplace environments [3].

The research described in this paper focuses on adaptation of modelling approaches, already applied in various other domains [2,7] in order to be used in the design and evaluation of open problem solving environments, the most notable adaptation being modelling of erroneous or unexpected user behaviour and development of combined models of multiple user behaviour.

The effort to apply model based design and evaluation methods has produced a design and evaluation framework and a number of tools that facilitate interaction modelling process in this domain. Since development of the models is a tedious process, special emphasis in this paper is provided to the description of a tool , Cognitive Modelling Tool, (CMT), developed to facilitate the modelling process, together with preliminary findings from the evaluation study of an open problem-solving educational environment, in which CMT was engaged.

DESIGNING FOR LEARNING AND USABILITY

While there seems to be wider acceptance of the importance of open environments in learning, design of such environments is not an easy task. Additionally there is a growing concern relating to the lack of methodologies and tools supporting their design and evaluation [5]. Students using these environments can solve problems in different ways, and in the process they can also make mistakes. From the constructivist perspective of learning, it is exactly these mistakes that can be treated as opportunities to explore new concepts [1]. So definition of usability in this context has to be adapted in order to achieve learnability, considered as the prime objective of educational environments. Squires and Preece [9], propose a heuristic evaluation framework, extended under the socio-constructivist learning perspective to ensure that usability features are not considered at the expense of educational issues or vice versa. However these heuristic evaluation techniques often fail to examine details of interaction and to model the process in a detailed enough way in order to become useful in system design. Some of these limitations can be addressed by modelling approaches which have been extensively used in HCI research and practice in order to build representations of users cognition, interaction, goals and task execution.

Cognitive modelling techniques, such as GOMS, model execution of *routine* tasks by *ideal* users. During the research reported here, we use task modelling as a core technique in evaluation and redesign of open problem-solving educational environments which do not bear these characteristics. Model-based design approaches involve incremental application of task modelling techniques, starting from the expected user task model, developed during the initial design and subsequently comparison of this model with the observed user's behaviour during field evaluation. The CMT tool, presented here, facilitates this process. The first findings of the proposed methodology, as derived from the evaluation of an open problem-solving environment, are discussed in subsequent sections.

MODEL BASED DESIGN

The proposed modelling approach is used to describe the interaction between the student and the learning environment and can be used to drive both the evaluation and design according to the following steps: During the requirement specification phase the designer, based on theories of learning, domain knowledge, requirements analysis, the expected users' characteristics and the social environment in which the system will operate, defines the primary designer's view of the users' task model (DTM). This model represents the way the designer expects the student would interact with the problem solving environment in order to accomplish typical tasks.

This process can be carried out using the *Cognitive Modelling Tool* discussed in the following section. Expected high level *goals* of the student are defined, further decomposed into sub-goals necessary to accomplish the high level goals. Generally, in an open problem-solving environment the individual students approach a given problem in different ways. Some of these ways should be reflected in the designer's task model, which is usually quite rich, containing many OR-related, alternative goal hierarchies. Special effort must be given to ensure that the proposed way of interaction embodies the desirable knowledge that the system should deliver, thus this model should be designed in co-operation with pedagogical experts.

The *field evaluation* phase serves to validate original designer's assumptions against actual student task execution. This field study may take place in a usability laboratory or in a school environment. Participants of the experiments must be representative typical students solving typical problems. The individual user's model reflecting her conception of the system is constructed by observing the individual student's interaction at the keystroke level (e.g. through log files) Also student's views about missing functionality of the systems can be detected. For each student, $n=t*s$ models are built, where t=number of high level tasks undertaken and s is the number of different strategies used to solve the given problems.

The user deviations captured at the keystroke level are mapped at the task level during this phase. If a usability-related mistake is detected then the task where the wrong interaction occurred is specially annotated with a detailed description and a grading of the severity of the problem. If an unanticipated problem solving strategy is observed, this is added incrementally to the original designer model. Remarks on additional functionality, required to support these unforeseen interactions, are also made. Conceptually wrong problem solving strategies can also be included in the student's interaction model, since they are often useful for educational purposes or can be used as typical misconceptions to be tackled by the system.

Following this process, a combined User Task Model (UMT) is derived which contains all the pupils' solution strategies detected and analysed. The purpose of this process is to unveil new ways of interaction that the original designer model could not support and eventually diminish the gap between the designer and the user models. After this analysis the original DTM is revised and augmented, which can drive the modification of the original design. Special effort should be given here, not to eliminate erroneous or incomplete task solving approaches which could be rich in learning value. Instead, analysis of student models should identify unrecoverable interaction errors, e.g. deadlocks in dialogue, and address them. In general, this augmented designer's model deals more effectively and in new diverse ways with the open educational environment redesign.

This approach, has been applied in the redesign and evaluation of the open problem solving environment, C.AR.ME (Conservation of Area and its Measurement) [4] This is an open problem-solving environment designed to support 12-14 year pupils to explore and perceive geometric concepts, and specifically the conservation of area and area measurement. The evaluation process took place in a school computer laboratory. Thirty (30) pupils, aged 13-14 took part in the experiments.. Pupils were given two typical "conservation of area and area measurement" tasks. The first task involved transformation of a non convex polygon to another one with equal area using all possible ways and the second task involved comparison of this polygon to a square non easily comparable by 'eye'. Each one of the students interacted individually with the software demonstrating various approaches of solving the given tasks.

THE COGNITIVE MODELING TOOL

The *Cognitive Modelling Tool (CMT)* has been developed to carry out the task modelling process described above. Rich designer and user task models are structured in a hierarchical way, as shown in figure 1.

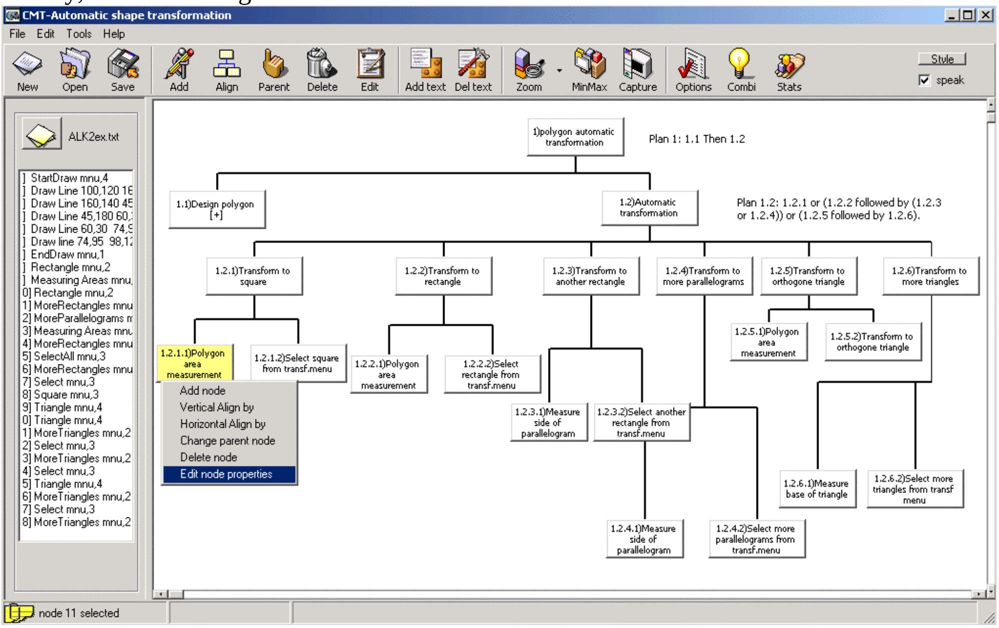

Figure 1: *Cognitive Modelling Tool : The main task model building space. Part of the designer model (DM) of C.AR.ME.*

CMT uses a direct manipulation approach for editing and modifying the graphical representation of a task hierarchy. Any specific node represents a task relating to a user goal. The sub-tasks which serve to accomplish this goal are associated to the node. These subtasks can be related through a specific plan involving OR, AND logic operators. This plan is usually shown next to the specific high-level goal. Additional information relating to comments on the goal accomplishment or deviations detected, either syntactic or semantic ones, can be attached to the node, as shown in task 1.2.2.1 of figure 2 (properties box).

Both textual and graphical notations can be used for task representation. This depicts an additional advantage that usage of the CMT tool offers : it can support communication across a design team (often consisting of designers from different disciplines and backgrounds) because of its ability of exporting task models in various possible representations: tree view, sequential view and structured report view produced automatically, all consistent with each other thus supporting the design process.

If the analysis is focused on the time required to accomplish a task, time related information can also be stored on each task and appropriate calculations regarding the time required for a task to be accomplished can be carried out automatically. Display of a keystroke log file, shown on the left of figure 1, next to the corresponding task model structure is supported. So both low level interaction details(keystrokes) and cognitive goal hierarchies are displayed simultaneously to the user of CMT. The possibility of dragging an event of the log file to the goal structure, which results in the introduction of a new node, filled with the action description, automates the process of goal structure building.

The task models are stored into a relational database, grouping the various systems analysed, with additional identification information (designer's model, revised designer's task model (DTM) or user's task models (UTM)). Also quantitative analysis procedures are supported to extract useful metrics related to the analysed tasks, such as number of keystrokes required to achieve a specific goal or sub-goal, mean time required and interaction complexity of a specific user model compared to primary designer's expectations or to the revised and adapted model.

A novel and time-saving functionality of the CMT tool is its ability to automate synthesis of task structures already stored in the database. Using this feature, various sub-goal structures can be combined or temporarily hidden away in a task model, according to the degree of detail required in the context of a particular study. Additionally, since task models are stored in a relational database schema, through database querying interesting comparisons can be made relating to various aspects of modelling, such as execution times and frequency of usage patterns, both in absolute or relative values (e.g. compared to other task structures). In CMT the evaluator or the designer can select parts of a task structure representing a specific problem solving strategy, which can be stored for future reference or comparison with other users' strategies. Additionally, CMT supports storage of various users' characteristics, such as age, grades and gender, so that further analysis can be supported, focused on them.

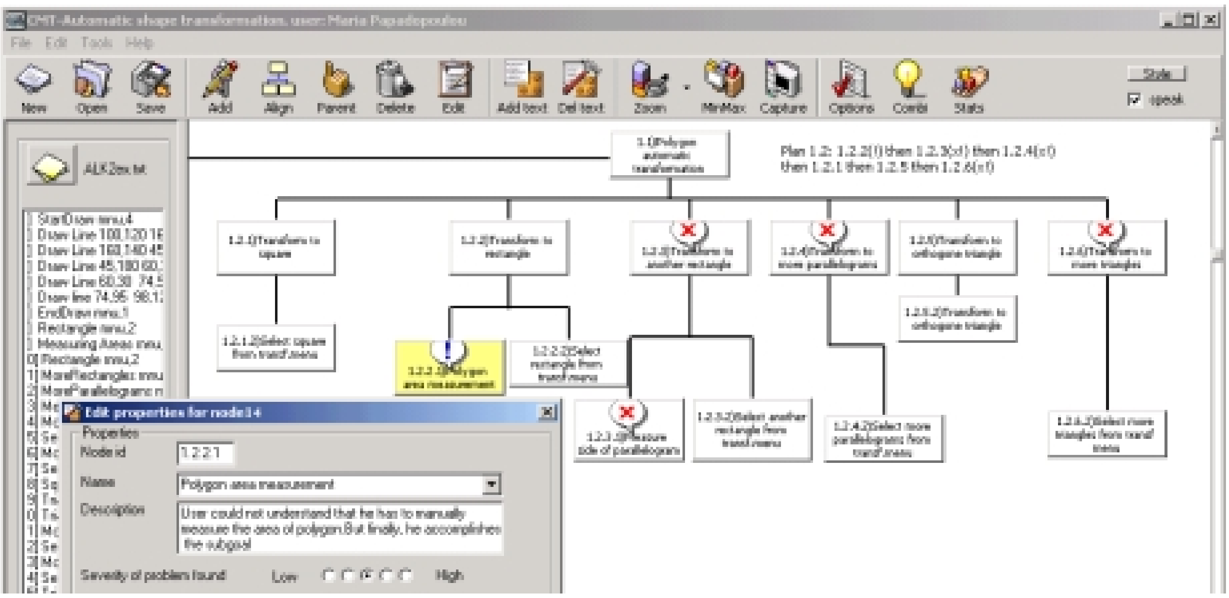

Figure 2: *Extract of User Task Model. sub-goal: polygon automatic transformation. Several tasks were marked as unaccomplished.*

An *individual user task model* can be annotated with comments relating to task execution. A special notation has been defined, extending the HTA plan notation [8], with tokens referring to user deviation from expected task execution. So token (!) marks a non-destructive syntactic deviation, while (x!) marks a not-completed task execution. Also introduction of new unforeseen tasks in a plan by a user are marked as {task-id}, as shown in plan 1.1 of Figure 3.

EXAMPLE

In this section, an example of C.AR.ME. evaluation is presented, demonstrating how the tool facilitated the analysis.

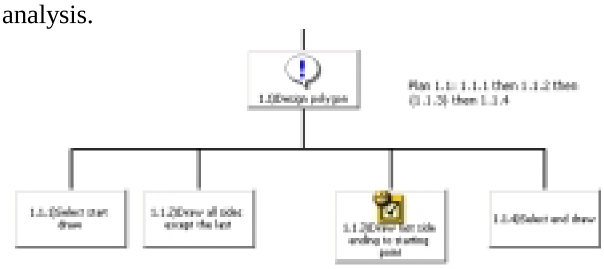

Figure 3: *User task analysis of a sub-goal identified during interaction with C.AR.ME.*

The presented task involves creation of a polygon and generation of geometric equivalent shapes of the same area that can be squares, rectangles and triangles. In C.AR.ME. this can be achieved using many tools, like gridlines, measuring units or automatic transformation of a reference shape. In the described example a pupil has opted for this last approach. Three specific cases of observed deviations of the user task execution in relation to the DTM, shown in figure1, are presented.

a)The *Reference polygon drawing* task (1.2.1) has been originally designed as follows: The pupil is expected to draw all the sides of the polygon except the last one and then to select "*end draw*" to complete the polygon. From the log file, it is deduced that the pupil attempted to complete the polygon by drawing the last point of the final segment close to the starting point, subsequently selecting the appropriate command "end draw". This was marked as a low severity syntactic deviation in the corresponding user task model. The additional task {1.1.3} was included in the task execution plan, as shown in Figure 3. A remark was also made that it is desirable that the C.AR.ME system should interpret drawing the end of a line near to the starting point, as an attempt to complete the polygon, thus providing more support to direct manipulation.

(b) The second misconception detected was related to the fact that in order to automatically produce the geometric equivalent shape, the user was prompted to manually measure the area of the original polygon. Demanding this manual activity in the frame of an automatic transformation task created confusion to the pupil. A comment was added to sub-task 1.2.2.1 in the UTM, shown in figure 3, describing the observed user behaviour and the relevant annotation mark (!) was assigned to the task execution.

(c) Subsequently, when the pupil attempted to create more equivalent shapes the system prompted her to measure one side of a previously drawn shape, in order to use it as a base. This does not clearly relate to the task objective. The result was that the user could not carry out the task and started trying different ways of interaction ending in a deadlock. Thus, the tasks 1.2.3, 1.2.4, 1.2.6 were marked as unaccomplished in the UTM, as shown in figure 3.

CONCLUSIONS

The CMT tool, developed to support the proposed modelling approach captures the details of student interaction with the problem solving environment both at the cognitive level as well as the keystroke level. This tool supports a novel approach of design and evaluation of open problem solving environments. As demonstrated in the described example, detailed analysis of students' problem solving behaviour through building of individual student task models could be carried out. Aggregation of these models in a single revised designer task model, can lead to a bottom up re-design phase following the original top-down design specification phase.

The observations made during this phase can be very rich, and their capturing and classification under the syntactic and semantic re-design perspectives, following this approach, have defined a framework for iterative design of open learning environments. Additionally, it has been proved that model-based approaches in general, permit deeper understanding of the nature of tasks during their decomposition and facilitate consistency control across task structures, so learning environments could benefit from a model-driven evaluation and design too. Additionally, this approach can be integrated with a constructivist perspective of learning evaluation process [11] as it supports the study of the development of the individual pupil strategies.

Also the CMT can be used for quantitative and qualitative evaluation of the problem solving process at both the individual student level and at the group level. By modelling the user interaction and problem solving strategies and judging them against the original model, as provided by educational experts, validation of the solutions provided by the problem solvers is facilitated. Moreover, this approach supports the study of each individual pupil strategies in relation to the general student model constructed by taking into account all pupils' strategies that participated in a particular study. Open problem-solving educational environments in general suffer from the lack of adequate learning evaluation mechanisms, since identification of correct solutions is often a tedious process.

Some shortcomings of the proposed approach relate to the difficulty of inferring the cognitive model of the user from the keystroke behaviour. This problem could be tackled by using adequate field evaluation protocols that involve deep interaction of evaluators with the user during the field study (e.g. think-aloud protocol, interviews etc). Also an automated strategy classification technique using Bayesian belief networks [10] has been proposed in order to classify click stream data, thus facilitating the student goal inferring process. However more research and experimentation is required in this direction.

ACKNOWLEDGEMENTS

Financial support by GSRT, under project PENED 99ED234, is acknowledged.

BIBLIOGRAPHY

1. Cobb, P. (1991). Reconstructing Elementary School Mathematics. *Focus on Learning Problems in Mathematics*, 13,(2), 3-32.
2. Gong R., Kieras D. (1994). *A Validation of the GOMS Model Methodology in the Development of a Specialized, Commercial Software Application*, Proceedings ACM CHI 1994 (Boston Massachussets, April 24-28), pp. 351-357.
3. Inkpen K. (1997). *Three Important Research Agendas for Educational Multimedia: Learning, Children, and Gender.* AACE World Conference on Educational Multimedia and Hypermedia 97, Calgary, AB, June 1997, pp 521-526.
4. Kordaki, M., Potari, D. (1998). *A learning environment for the conservation of area and its measurement: a computer microworld*, Computers and Education, 31, pp. 405-422.
5. Kordaki, M., Avouris N, (2001). *Modelling in Design and Evaluation of Open Learning Environments*, Computers and Education (submitted).
6. Paternó F., Ballardin G., (2000). *RemUSINE: a Bridge between Empirical and Model-based Evaluation when Evaluators and Users are Distant,* Interacting with Computers, Vol.13, N.2, pp. 151-167.
7. Paternó F.,Mancini C. (2000). *Model-based design of Interactive Applications*, ACM Intelligence,winter 2000,pp.27-37.
8. Shepherd, A. (1989). *Analysis and training in information technology task*. In Diaper, D.(ed.); Task Analysis for human computer interaction, Ellis Horwood Limited, pp.15-55.
9. Squires D., & Preece, J., (1999). *Predicting Quality in Educational Software: Evaluating for Learning, Usability and the Synergy between them,* Interacting with Computers, 11 (1999), pp. 467-483.

10. Tselios N., Maragkoudakis M., Avouris N, Fakotakis N., Kordaki M., (2001). *Automatic diagnosis of student problem solving strategies using Bayesian Networks*, 5th Panhellenic Conference in Didactics of Mathematics and Informatics in Education, Thessaloniki, 2001 (in Greek).

11. Von Glasersfeld, E. (1987). *Learning as a constructive activity*. In Janvier, C. (Ed.) Problems of representation in teaching and learning of mathematics London: Lawrence Erlbaum associates, pp. 3-18.