Tools for task-based interaction and collaboration analysis

Nikolaos Avouris[1], Georgios Fiotakis[1], Nikolaos Tselios[1], Vassilis Komis[2]

[1]Electrical & Computer Eng. Dept, HCI Group, [2]Early Childhood Educ. Dept.,
University of Patras, 26500 Rio Patras, Greece
{ N.Avouris, Fiotakis, NiTse }@ ee.upatras.gr, komis@upatras.gr

Abstract

An innovative framework of interaction and collaboration analysis is proposed, jointly with tools to support the process. The proposed framework is based on a *collaboration analysis* first and *individual task analysis* subsequently, approach. The objective of this framework is to facilitate understanding of the group's and individual user's tasks and goals and associate the artifacts used with usability problems. An innovative aspect of the framework is the association of tasks to artifacts (tools) engaged by the users during the activity. The typical use of this framework is in interactive systems evaluation and design. The framework and the tools functionality are described in the paper. The framework is inspired by the *Activity Theory* perspective, which recognizes the importance of artifacts, actors and the context in which an activity takes place.

1 Introduction

Tools and techniques to support interaction and collaboration analysis have been proposed in the frame of usability evaluation studies for many years now (Dix et al., 1998, Nielsen 1994). These techniques involve analysis of data that are collected from field studies in various forms. Stream media like audio and video, logfiles, as well as notes and comments of observers are used in these studies. Discrete data items, like files containing solutions to problems, drawings, etc. may also be used. All these data need to be correlated and processed in order the researchers to extract useful patterns of behaviour of the actors involved, identify usability and conceptual flaws in the design of the artifacts used and evaluate the approaches that have been pursued. This analysis process has become tedious, since the high volume of data has made it more time-consuming and complex. The need for adequate tools to support the analysis has therefore increased recently. A framework of interactive systems analysis is proposed in this paper, which involves two complementary views over an activity involving multiple actors, supported with relevant tools:
(a) The first one concerns analysis of collaboration, which involves collection of field data, annotation of these observations and building of an abstract description of observed collaborative task execution. A tool has been built for annotation of data and building of abstract multi-level annotated views. This tool (Collaboration Analysis Tool, ColAT) bears interesting characteristics among which the support for various annotation schemes, the capability to commend and annotate at various levels of abstraction a sequence of events, interrelation of multiple media (video, audio, log files, snapshots) to the multilevel annotation and to the cognitive structures built through the second view.
(b) The second view is a cognitive one, which involves building of typical task structures, as anticipated by the designers of the artifact (e.g. the computer tools that are used in the activity), subsequently relating the observed task execution to the anticipated model by individuals who participate in the activity. A tool has been built to support this task-based approach, the Cognitive Modelling Tool (CMTool), also described in (Tselios & Avouris 2003). Analysis characteristics of

task models are stored in the CMTool database, together with qualifying information. The possibility of analysing further the observed system usage according to a number of dimensions permits evaluation of the artifacts both in terms of usability and effectiveness.

The proposed framework and analysis tools have been applied in non-routine task domains (Tselios & Avouris 2003). The findings reported indicate the effectiveness of this structured technique in identifying usability and interaction design problems. In the reported studies, we focused on analysis and evaluation of collaborative tasks, involving a number of individuals and artifacts engaged in problem solving either at a distance or in the same place. The ethnographic methodological approach used in the evaluation studies of this nature proved to be compatible to the proposed theories underlying the framework and the tools.

2 Analysis of observed collaborative activity

The first phase of analysis involves collaboration analysis study. A new integrated environment of analysis, the *Collaboration Analysis Toolkit* (ColAT), which integrates multiple sources of behavioural data of multiple logging and monitoring devices is used in this phase. The main emphasis of the ColAT environment is on the analysis of situations involving more than one actors. Special attention has been put on scenarios of synchronous computer-supported collaborative problem solving, in which the actors are spatially dislocated, a factor which imposes additional complexity in the analysis task.

The most important phase of analysis relates to the interpretation and annotation of the collected data, as well as generation of aggregate data of interpretative nature. An innovative feature of the ColAT approach is the support for creation of a multi-level structure that describes and interprets the logfile events. In figure 1 the concept of the multi-level logfile is shown.

The original sequence of events contained in the master logfile is level 1 (*events level*) of this multilevel structure. The keystrokes or raw observations are included in this level. An example is the event "User X selects option Y from the menu" or "User Z said …."in case of a dialogue event. A number of such events can be associated to an entry at *level* 2 by the analyst. Such an entry can have the following structure:

< *ID, User, entry_type, comment* >

where *ID* is a unique identity of the entry, *type* is a classification of the entry according to a typology that has been defined by the researcher, followed by a textual comment or attributes that are relevant to this type of task entry. Examples of entries of this level are:" User X inserts a link in the model", or "User Y contests the statement of User Z". In a similar manner the entries of the higher levels are also created, which describe the activity at the strategy level as a sequence of interrelated goals of the actors involved.

An implication of this approach is that the associated stream media are related to this multi-level view of the activity and therefore the user of ColAT can decide to view the activity from any level of abstraction he/she wishes. This approach results in a powerful analytical tool, since the possibility of viewing a process from various levels of abstraction supports its deeper understanding and interpretation.

A "*ColAT project"* is stored in a database to facilitate processing and navigation of the source data and annotations. The integrated logfile can be exported in XML form to other applications and data processing tools for further analysis. The main activity of this phase involves creation of the higher-level logfile entries. In these higher levels of the logfile the typology of the Object-Oriented Collaboration Analysis Framework (OCAF), see (Avouris et al. 2003a), has been used. This framework is particularly suitable for analysis of collaborative activity, which involves interleaving of actions and dialogue. OCAF puts emphasis on the objects of the jointly developed solution. Every object is assigned its own history of events (actions and messages) related to its existence, as a sequence of events according to the following functional types: I = Insertion of the item in the shared space, P= Proposal of an item or proposal of a state of an item, C= Contestation of a proposal, R= Rejection / refutation of a proposal, X= Acknowledgement/ acceptance of a proposal, T= Test/Verify using tools or other means of an object or a construct.

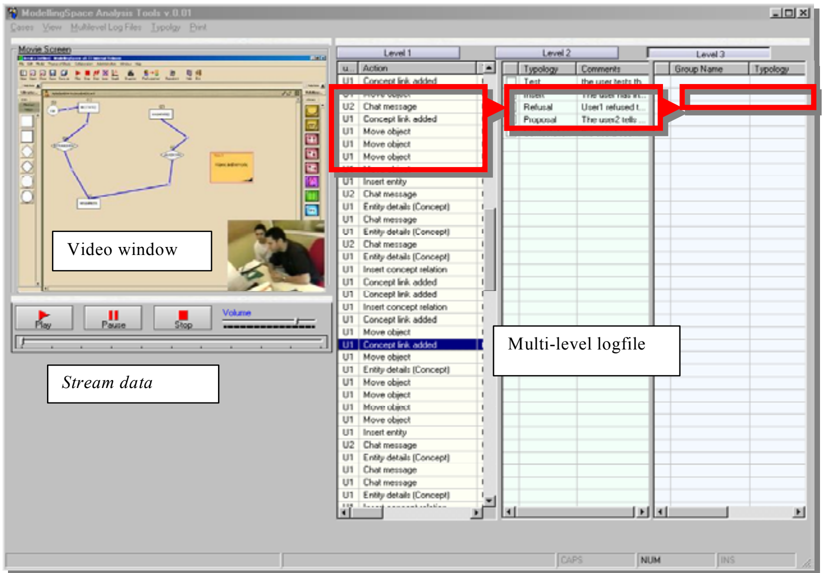

Figure 1: Overview of the ColAT data navigation environment

As an example of an OCAF event, the introduction of a new object in the shared space, is indicated as *Object (X)*= I_{U1}, i.e. User 1 inserted the object (X) in the shared space.

The ColAT environment that supports navigation of the constructed multilevel logfiles is shown in figure 1. A video window permits viewing of streaming data in association to selected events in any level of the logfiles. There are different modes of use of this environment: In the first mode, *navigation is controlled through the video*. When a position of the video file is selected, the corresponding event of the log hierarchy that the video is related to, is highlighted. In the second mode, *navigation is controlled from the logfiles*. In this case the user can select any event in the first level of the log file and the video starts from that event onwards.

The user can hide the levels of abstraction he/she wishes to ignore, thus defining the desired view over the field data. The ColAT navigation tool has been proven particularly useful in analysing the data of reported experiments (Avouris etal. 2003b), following the OCAF framework.

3. The task analysis of individual actors

During the second phase, each individual actor is studied in relation to the tasks she undertook and the goals she attempted to accomplish. The task analysis method adopted is the Hierarchical Task Analysis (HTA), proposed by Shepherd (1989), accordingly modified, as discussed here. The intention is to build a conceptual framework reflecting the way the user views the system and the tasks undertaken in order to accomplish set goals. So the main objective of our analysis is to reflect on the observable behaviour of users and identify bottlenecks in the human-system dialogues rather than explicitly understand the internal cognitive processes as one performs a given task. So we include in our analysis, even incorrect problem solving approaches, caused

either by limitations in the provided tools design or by conceptual misunderstandings related to domain knowledge and use of available tools. These errors may lead to not satisfactory solutions to a given problem, but often do contribute to better and deeper understanding of concepts.
Through the proposed task analysis technique, the typical or expected use of the software environment by each individual user is reduced, to a sequence of tasks. However, this task analysis can be achieved if the right level of abstraction is used, as fed from the previous phase of the study. Kirwan and Ainsworth (1992) and Boy (1999) show methods of accomplishing this.

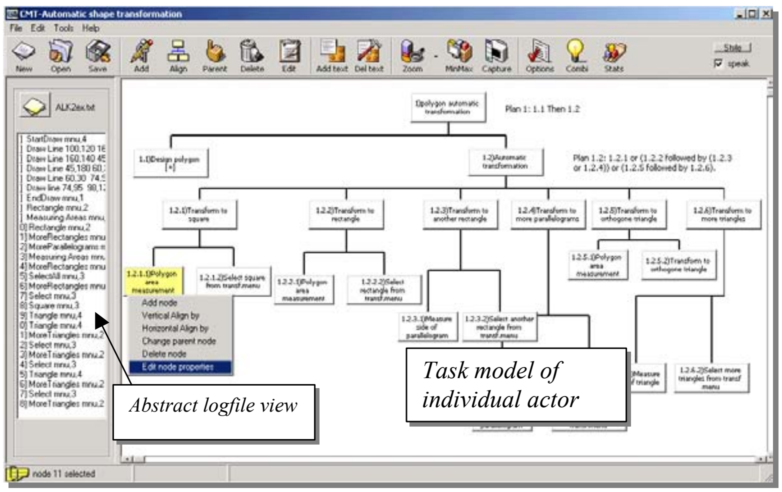

Figure 2: The Cognitive Modelling Tool (CMTool) Environment

One important aspect of our analysis framework is the classification of observed unexpected or incorrect user behaviour, see Tselios et al. (2002) and Tselios & Avouris (2003). Through the application and analysis of this technique, we have identified and classified five main categories of errors: *Severe syntactic error, Minor syntactic error, Conceptual error, Inconsistency, Design principle violation*. These errors can be identified during observation and analysis of problem-solving activity during this phase. Also in *plans* (Shepherd, 1989), simple expressions are included such as (!) and (x!) to denote subtasks containing errors, and { } to indicate occurrence of unforeseen tasks according to the original designer model.
Display of an annotated log file at selected level of abstraction, shown on the left of fig. 2, next to the task model structure, is supported. So both interaction details and cognitive goal hierarchies are displayed simultaneously to the user of CMTool. Task models are stored in a relational database, grouping the various tasks analysed, with additional identification information (designer's model or revised designer's task model (DTM) or user's task models (UTM)). In addition, quantitative analysis tools are supported to extract useful metrics related to the analysed tasks. Examples of these metrics are the number of keystrokes required to achieve a specific goal or sub-goal, the mean time required and the interaction complexity of specific user model, compared to the original designer's expectations or to the revised and adapted model.
In CMTool, the evaluator 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. In addition, the possibility to analyse system's usage in five dimensions is a contribution of the tool to the evaluation process. These are: (i) High level tasks, (ii) users, (iii) specific strategies, (iv) tools used, (v) usability problems detected. This process is carried out through a visual information query environment. Field experiments could be analysed across any possible element of the five different dimensions discussed (e.g. "Show all encountered problems

related to tool X", etc.). Complex analysis can be carried out according to any of these dimensions, supporting study and analysis of encountered problems.

4 Conclusions

This paper describes the main functionality of tools to support multilevel analysis of field data collected during evaluation studies of group collaboration. Both *ColAT* and *CMTool* have been recently applied in evaluation studies of group problem solving activities. The theoretical underpinning of this approach is Activity Theory (Kaptelinin etal. 1999). This is a conceptual approach, that considers as the unit of analysis the activity, consisting of a subject (an individual or group), an object or motive, artefacts, and sociocultural rules. Understanding thus human activity requires a commitment to a complex unit of analysis. The multi-view approach proposed covers this aspect. In the proposed framework we move from the group level analysis to the individual human-computer interaction study in a smooth way. The multi-level annotation scheme described here permits change of point of view and relates the observational data to the annotations. These annotations can comply with a typology imposed by a methodological framework, like the OCAF scheme used in our example. The framework permits shifting from bottom up annotation of group activity data to top-down task level description of the observed human-computer interaction

The concepts and tools discussed here are relevant to researchers who are involved in analysis and evaluation of complex collaboration-support activities, in design and evaluation of new tools and in support of users' meta-cognitive activities.

Acknowledgement

Partial funding of project IST-2000-25385 ModellingSpace, by the EC is acknowledged.

References

Avouris N.M., Dimitracopoulou A., Komis V., (2003a). On analysis of collaborative problem solving: An object-oriented approach, Computers in Human Behavior, 19 (2), March, pp. 147-167.

Avouris N., Komis V., Margaritis M., Fiotakis G., (2003b). Tools for Interaction and Collaboration Analysis of learning, Proc. Conf. CBLIS 2003, July, Nicosia, Cyprus.

Boy, G. A. (1998). Cognitive Function Analysis. Ablex Publ., Greenwood, Westport, CT, USA.

Dix A., Finlay J., Abowd G, Beale R., (1998), Human-Computer Interaction, Prentice Hall

Kaptelinin, V., Nardi B., Macaulay C., (1999). The Activity Checklist: A Tool for Representing the "Space" of Context, Interactions, July, 27-39.

Kirwan, B. and Ainsworth, L.K. (1992). A Guide to Task Analysis. Taylor & Francis, London.

Nielsen, J., (1994). Usability inspection methods. In J.Nielsen, R.L. Mark (Ed.), Usability Inspection Methods, John Willey, New York.

Paterno', F. (2000) Model-based design and evaluation of interactive applications, Springer Series in Applied Computing, Springer-Verlag, London.

Shepherd, A. (1989). Analysis and training in information technology task. In Diaper, D.(Ed.) Task Analysis for human computer interaction. Ellis Horwood Limited, 15-55.

Tselios N., Avouris N., (2003), Cognitive Task Modeling for system design and evaluation of nonroutine task domains, E. Hollnagen (ed.) Handbook of Cognitive Task Design, LEA, 307-332.

Tselios N., Avouris N., Kordaki M., (2002), Student Task Modeling in design and evaluation of open problem-solving environments, Education and Information Technologies, 7:1, 19-42.