Interaction Analysis with dedicated logfile analysis tools – a comparative case using the PAnDiT tool versus manual inspection

Andreas Harrer\textsuperscript{a}, Andreas Lingnau\textsuperscript{b}, Martina Bientzle\textsuperscript{b}
\textsuperscript{a}KU Eichstätt-Ingolstadt, \textsuperscript{b}KMRC Tübingen, Germany
andreas.harrer@kuei.de, \{a.lingnau, m.bientzle\}@iwm-kmrc.de

Abstract

This article presents an interaction analysis tool and its usage for a collaborative learning situation that has been analysed manually before. We want to contrast and discuss the use of dedicated analysis tools compared to the current practice of evaluation using standard software or no computer tools at all.

1. Introduction

Interaction analysis has been a major research strand in the area of Computer-supported Collaborative Learning (CSCL) in recent years. This is shown by the elaborate chat analyses \cite{1}, rating and analysis of video-captured collaboration \cite{2} and analytical frameworks \cite{3}. Yet, the practice of researchers when analysing data is currently mostly to use standard tools, such as spreadsheets and video players with little annotation capability. Our intention is to show that dedicated interaction analysis tools can support the researcher in her / his work and substantially reduce the effort of performing the analysis.

2. The PAnDiT analysis tool

Most CSCL systems produce logfiles / protocols of learner and system actions during collaboration automatically. In distributed scenarios, i.e. where the students do not work at the same computer, the propagation of events is performed anyway, which makes logging an add-on easy to implement. Thus, usually the data about direct interaction of learners with the system is readily available for analysis. This was the starting point to develop a dedicated logfile analysis tool for CSCL settings. Our main criteria for development were to be flexible with the data sources that can be analysed and to empower the researcher to formulate rules to detect specific situations in the collaboration without asking explicit programming knowledge from the researcher. We will describe shortly the functionality of the current version called Pattern Analysis and Discovery Tool (PAnDiT, the Sanskrit word “Pandit” meaning “scholar”).

The tool accepts structured logs in a format (CoLoForm = Collaboration Logfile Format) that has been agreed on by several international research labs during the European Kaleidoscope network of excellence. Besides this flexible XML-based format that has been used in several collaborative applications and analysis tools, PAnDiT also offers an option to transform logfiles to this given a proper XSL-transformation script. Compliant logfiles are read in and generic statistical information is produced, such as activity over time, activity of individuals over time, distribution of activity between all participants, and a spider diagram to inform about specific foci of work of each group member. The overview window of the PAnDiT tool is shown in Figure 1.

After this first overview the researcher can define

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Overview of logfiles with PAnDiT}
\end{figure}

the questions she / he is interested in by means of rules that are checked on the logfile data. Besides the mere counting of events (e.g. a rule “filter the logfile and give as a result all deletion actions of Student1”) the rules allow also to analyse more complex sequences of
actions by combining constraints between the different steps (e.g. “filter the log for all sequences where two students exchanged a pair of chat messages and directly after that one of these created an object”). The definition of rules is supported by a drag-and-drop approach that adds the dragged log event to the sequence of actions. For each action the attributes (such as action type, object type, object id, user performing the action etc.) can be further specified either using concrete values or variables: using the variables in several actions allows the constraints between actions, such as “the same student created and deleted the object” or “a different student deleted the object after the first created it”. While usually the computational (Prolog programming language) level is hidden from the user for reasons of intuitive use and user-friendliness, experts can change to a plain Prolog representation and modify the rules directly for even more expressive rules. On execution of a ruleset (a collection of rules), the “hits”, i.e. parts of the logfiles that comply to a rule, are presented in a hierarchical (for each rule) and tabular format (cf. Figure 2 left) as well as in a timeline view showing time in the x-axis, the different users in the y-axis (cf. Figure 2 on the right). Hits are presented in the timeline as dots (for individual actions) or polylines (for rules consisting of several actions).

Once a researcher has codified her / his research question as a ruleset, he can store it and use it on other logs as well; yet, the automated analysis of a full research corpus (e.g. all logfiles of a study) can be easily supported with this approach, too. By choosing a ruleset and a folder all the logfiles contained in the folder are analysed and a table is generated that shows how many hits each rule produced. Directly from the table the researcher can “zoom in” on the detailed analysis of any of the logfiles. Before we present how the PAnDiT tool has been used for interaction analysis, we will shortly describe the context and learning scenario that has been analysed.

3. The learning setting to be analysed

Verbal communication, particularly the ability to give directions and understand them, is a key not only for learning but also for everyday life. One of the main objectives of schools for pupils with cognitive disability or learning difficulties is to prepare them to manage their everyday life on their own as much as possible. In a recent project [4] collaborative settings were used to support pupils with cognitive disability to develop coordination and communication skills. In our case the students have to agree on the position of objects they have to place in a shared workspace by means of a specific confirmation protocol [5]. In the specific scenario to be analysed the task for the pupils was to place furniture objects into the appropriate rooms of a house. In the floor control condition every object was assigned to one pupil explicitly for placement and the other pupil could confirm the position or reject it via a traffic light metaphor (green for acceptance, red for rejection). On rejection the first student has to propose another placement and the same agreement procedure starts over until acceptance is reached or a specific number of rejections is reached and the object is dropped. In the condition without floor control both pupils can move the object after the initial placement. After each placement the other pupil can either accept the position of move it to another, which is implicitly considered a rejection and a counterproposal. One of the research questions explored in the study is if the floor control condition affects level of activity and balance of activity between pupils. This is especially important because in earlier experiments without coordination support a dominance of one pupil was apparent [4].

4. Manual and tool-supported analysis

In a first evaluation a manual analysis method was used using video inspection and the replay functionality the collaboration tool provides. The replay mode uses the logs that have also been used for the tool-based analysis described later on. 20 students distributed over 5 dyads for both conditions participated in the study. The logfiles captured by the collaborative application have been made compatible with the CoLoForm data format in a recent project and
thus were directly usable for the interaction analysis. A first ruleset for the automated analysis was produced in a phone conference (less than 2 hours) between the authors (the experimenters and the PAnDiT expert, that have not been directly working on this study together before this analysis) and has been slightly modified to follow closely the method for the manual analysis.

The relevant attributes contained in the logfiles that are required for an operationalization as rules are ObjectType, ActionType, User, RejectCounter, ConfirmationFlag 1 and 2. RejectCounter is an internal variable that stores the number of rejections for an object to allow the cancellation of an object in case the pupils cannot agree (in the study the threshold was put to 3 rejections). The ConfirmationFlag attributes show the current status of agreement of pupil1 resp. pupil2. To compute the balance between pupils activity we had to define rules for “moving an object” and “rejecting a position” for each of the pupils. Because of the specific character of the floor control where only one pupil can move objects, the sum of moves and rejections has to be counted to make the two conditions comparable.

The rule for rejections of pupil1 considers all events that are of type “modify: attributes” on a FloorControlPuzzle object. The essential constraints on the attributes are that the user attribute has to have the value “pupil 1” and the confirmation flag of pupil 1 is “false”. In analogy to that the rule for rejections of pupil2 is codified. Figure 2 shows the rule matches for one specific logfile where pupil1 rejected the position of the partner five times and pupil2 seven times, which shows at one glimpse the balance of rejections.

Similarly, the rules for “moving an object” search for all events of type “modify:move/resize” on FloorControlPuzzle objects. The constraints of having the user attribute filled either with “Schueler1” or “Schueler2” differentiate the rules for the pupils.

This ruleset was stored after the successful test with one logfile and used to analyse all the logs at once for the general view about balance. The comparison with the original manual analysis shows a good replication of the results: while manually the average balance of activity for floor control was given as 13.6 : 8, our computational rule-based analysis produced the result 14.25 : 7.5. The slight difference can probably be accounted by both the expertise in interpretation by the analyser (some episodes on video using the audio communication of the pupils might be interpreted as rejection even though formally the logfiles do not show this). While in completely manual analyses this is usually compensated by measuring inter-rater reliability, a computational approach could – in our opinion - also be used as a benchmark of objective comparison. We plan to substantiate our claims with larger datasets in further research, yet, having a small scale study as a first opportunity to test the approach was promising.

### 5. Conclusion and outlook

In this article we motivated the need for dedicated interaction analysis tools to support researchers in their evaluations and increase the efficiency of the analysis process. We presented the PAnDiT tool that was designed specifically for logfile analysis of collaborative learning activities. Using a flexible rule-based approach the researcher is supported in operationalizing research questions once, yet the codified rulesets can be applied for whole datasets over and over, also across different studies and contexts. With this re-usability we assume a substantial time and cost gain for the researchers. To validate our claim we conducted a re-analysis of a collaborative setting for pupils with cognitive disabilities. The manual analysis that required several weeks of reviewing and interpreting video-taped interaction was contrasted with the computational approach using PAnDiT with an estimated analysis effort of 4 hours (including the creation of rules and aggregation of results). Larger and additional datasets can be analysed afterwards with the existing rulesets without any additional effort.

### 6. References


