

Looking at interaction data in log files as instances of process models: The value of process mining methods for CSCL.

Peter Reimann, University of Sydney

Most log-file based interaction analysis methods treat the data in the log files as a sequence of events, as different from changes in the values of variables over time. But it is often not rationalized what the appropriate level of granularity is on which these events should be described. However, finding the right level of granularity is important for both conceptual reasons--as it reflects ontological and theoretical assumptions about the nature of interaction processes in groups--as well as practical reasons: computational log files analysis, in order to yield psychological and pedagogical relevant information, needs to be based on the appropriate level of data granularity. For instance, while it is technically feasible to track users' interactions with tools and group members on the level of mouse-clicks, describing events on this level is hardly informative when the objective is to capture knowledge building processes taking place in a semester long on-line course. For more on the conceptual issues related to event definition, see [1], and for a discussion of issues relevant for data mining, see [2].

In order to make the issue of event granularity more concrete, I suggest the following (heuristic) distinction between three levels: series, sequences, and narratives. These may be seen as points on a granularity dimension with *atomistic* event models on side and *holistic* ones on the other end. The notion of a sequence suggests a more holistic view of process than the notion of a series. For instance, when we speak of a decision making process in a group, we refer to a process that has a beginning and an end, comprises a number of sub-steps (events), and a number of constraints on the order of the sub-steps. However, a sequence does not have to have a plot-like structure, and does not have to convey all the details typical for a narrative. Hence, sequences can be seen as conceptualizations of process more granular than series, and less holistic than narratives. Intuitively, for a sequence (and a narrative) the *form* of the sequence matters somehow, while for a series all that matters is preserved in the information contained in immediately adjacent events.

I argue that the sequence level is particularly relevant for CSCL because on this level (and up) we can describe groups as *activity systems*, following the classification of change processes in groups suggested by [3]. They distinguish (a) developmental processes, which are inherent to the system; (b) adaptational processes “generated by the group's response to (actual or anticipated) changes in the embedding context” (p. 6); (c) learning processes, which are based on a group's experience and reflection thereof; and (d) the group's operational processes, actions and activities, which are hierarchically and sequentially related. A group as an activity system carries out (partially) planned operations in the course of working on projects with specific goals.

The question arises next how observed sequences can be grouped and classified. One way to do this is to look for *patterns*, for *typical* sequences. One way to find patterns is to use optimal matching algorithms based on a similarity measure for sequences such as the number of changes required to transform one sequence into another [e.g. 4] or to

cluster observed sequences in other ways [5]. Another approach for pattern identification is to rely on graphical representations and use visual cues to group sequences into clusters [e.g., 6].

Another way to look at sequences is to see them as *generated* by an abstract process - to treat observed sequences as *instances* of a *model*. This is in particular appropriate when the sequences in the log files can be expected to reflect structured group activities, such as resulting from scripted collaboration [7] or from project-based cooperation [8]. As we said before, in such cases, we can think of groups as *activity systems*—as entities that carry out their projects, and of a log file as containing at least in parts records of these structured (planned, coordinated) activities.

We suggest to use process mining methods to identify typical sequences. Process Modeling has roots in Business IT and theoretical computer science rather than research computing, but has a number of characteristics that make it interesting to add to the repertoire of process analysis methods in CSCL. (Note that we use capitals for Process Model and for Process Modeling in order to distinguish this specific approach from the general notion of process models--which can take many forms, amongst them Process Models.) Process Models are interesting conceptually because they describe processes holistically, incorporating a priori assumptions about the form a process and all its instantiations can take. This makes Process Models suitable to describe *designed* processes, with the design effecting process enactment through prescriptions (e.g., collaboration scripts) and/or through constraints built into the collaboration software (e.g. an argumentation ontology, or specific features in the user interface). Process Models are interesting furthermore for practical reasons as they can under certain circumstances be identified automatically from log data.

A Process Model in the meaning intended here is a formal model, a parsimonious description of all possible activity sequences that are compatible with a model. A defining characteristic of Process Modeling is that log file data are seen as being *generated* by a process (in general, this can be multiple processes, but to keep the explanation concise will speak of one process only) and that this underlying process can be modeled as a discrete event system. More precisely, the log file is interpreted as a sequence of activities that result from (typically) multiple enactments of a process; these enactments form the process *instances*. Since the events in the log file correspond to a more or less small number of activity classes, they can be described with a limited vocabulary.

Process mining algorithms come in a variety of forms. In the workshop, we will introduce the main categories as well as the underlying assumptions in more detail than is possible here. Here an example will suffice. [9] provide an example of applying process mining to chat data, using the HeuristicsMiner algorithm [10] in order to identify the common form of decision making processes as embedded in the chat logs. The result of applying this algorithm [with specific parameters set as described in 9] takes the form of a dependency graph depicted in Figure 1. The arcs on the right side of the boxes that point back at their own box indicate loops, meaning that statements of this type often occurred multiple times in a row. The numbers along the arcs show the dependency of the relationship between two events, as explained previously. The

second number indicates the number of times this order of events occurred. The numbers in the boxes indicate the frequency of this event.

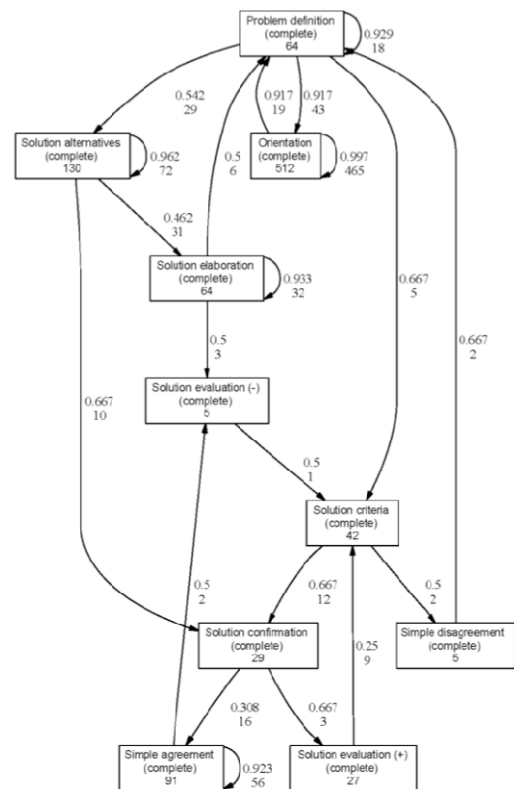


Figure 1: Dependency graph as a visual representation of a group decision making process

Space constraints do not permit us to go into the details of this specific model. Instead, let us mention some general points. It is important to note that the Process Model discovered from the (coded) chat transcripts and taking the form of Dependency Graph as displayed in Figure 1 is a model of *all* the (23) process instances occurring in the chat transcripts; it is an aggregation, or generalisation, of these observed instances, but not one using variables. Since the model is discovered using heuristics, the decision if newly observed decision instances (for instance, from a different group of students, or from the same group of students at a different time) is commensurate with the model can not be made in a deterministic manner (as is the case for Petri Nets), but would need statistical methods.

1. Abbott, A., *Event sequence and event duration: Colligation and measurement*. Historical Methods, 1984. **17**: p. 192-204.
2. Perera, D., et al. *Mining learners' traces from an online collaboration tool*. In *Educational Data Mining Workshop at Artificial Intelligence and Education*. 2007.
3. McGrath, J.E. and F. Tschan, *Temporal matters in social psychology: Examining the role of time in the lives of groups and individuals*. 2004, Washington, DC: American Psychological Association.

4. Abbott, A. and A. Hrycak, *Measuring resemblance in sequence data: an optimal matching analysis of musicians' careers*. *American Journal of Sociology*, 1990. **96**: p. 144-185.
5. Kaufman, L. and P.J. Rousseeuw, *Finding groups in data. An introduction to cluster analysis*. Wiley Series in Probability and Mathematical Statistics. 1990, New York: Wiley.
6. Suthers, D.D., *A qualitative analysis of collaborative knowledge construction through shared representations*. *Research and Practice in Technology Enhanced Learning*, 2006. **1**(2): p. 115-142.
7. Weinberger, A. and F. Fischer, *A framework to analyze argumentative knowledge construction in computer-supported collaborative learning*. *Computers & Education*, 2006. **46**(1): p. 71-95.
8. Zumbach, J. and P. Reimann, *Influence of feedback on distributed problem based learning*. Paper presented at the CSCL 2003 conference, June 15th to 18th, Bergen, Norway. 2003.
9. Reimann, P., J. Frerejean, and K. Thompson, *Using process mining to identify decision making processes in virtual teams (under review)*, in *International Conference on Computer-supported collaborative learning (CSCL2009)*. Rhodes/Greece. 2009, International Society for the Learning Sciences: Rhodes.
10. Weijters, A.J.M.M., W.M.P. Van der Aalst, and A.K.A.d. Medeiros, *Process mining with the heuristics miner-algorithm*. *BETA Working Paper Series WP 166*. 2006, Eindhoven University of Technology: Eindhoven, NL.