Using Process Mining to Identify Models of Group Decision Making in Chat Data

Peter Reimann, The University of Sydney, Building A35 Faculty of Education and Social Work, The University of Sydney NSW Australia 2006, p.reimann@edfac.usyd.edu.au
Jimmy Frerejean, Maastricht University, Faculty of Psychology and Neuroscience, Universiteitssingel 40, Maastricht, The Netherlands, Jimmy.Frerejean@psychology.unimaas.nl
Kate Thompson, The University of Sydney, Building A35 Faculty of Education and Social Work, The University of Sydney NSW Australia 2006, k.thompson@edfac.usyd.edu.au

Abstract. This paper introduces *process modeling and mining* as an approach to process analysis for CSCL. This approach is particularly relevant for collaborative learning that takes a project-based form, and is applied in this study to online chat data from teams working on a complex task. The groups differed in terms of the number of members and the amount of scaffolding aimed at group processes and task requirements. The models, produced using the HeuristicsMiner algorithm, showed that the group with fewer members that received more instruction in the task requirements had a more linear decision-making process than the group that received instruction in group processes, however neither were an example of a linear, unitary phase model. This approach has relevance both for CSCL research methods and for providing feedback to students on their decision-making processes.

Introduction

While an important feature of research on computer-mediated work and learning is that the researcher has access to detailed traces of the interaction between humans and machines, in an the case of CSCW and CSCL of the interactions between humans mediated by technology, analysing these data from a process perspective is still challenging. With process analysis, we refer to theories and methods that take the temporal nature of problem solving and learning into account. For group work and group learning, it means that the development of groups over time is taken into account. Temporality does not only come into play in quantitative terms (e.g., durations, rates of change), but *order* matters: Since human problem solving and learning is inherently cumulative, the sequence in which experiences are encountered affects how one learns and what one learns (Ritter, Nerb, Lehtinen, & O'Shea, 2007). This can certainly be generalized to groups, and to the communication and interaction processes that take place in groups in addition to learning. Each group has a history, and this history affects their activities and their learning (McGrath & Tschan, 2004).

The order of events has been carefully considered and theorized in conversation analysis (e.g., Schegloff, 2007), that is, for data that takes the form of talk, or talk-like communication such as on-line chat extending typically over seconds or minutes (Stahl, 2006). In studies where interaction and learning is distributed over multiple sessions (and perhaps multiple media, in particular for asynchronous interaction) and where log files are the main data corpora, temporal processes have been less well theorized and process analysis is less often practiced (although there are examples, e.g. Schümmer, Strijbos, & Berkel, 2005). Methodological challenges increase as the time intervals considered for analysis become longer. For instance, as time increases, non-controlled factors will come into play with a higher probability than is the case for short-term collaboration, and changes in group membership become more frequent, thus qualitatively changing the studied 'unit'. Non-linear changes will become more pronounced because of the self-sustaining feedback processes at work in groups over time (Arrow, McGrath, & Behrdal, 2000); that is to say, small differences can have large effects. Development in groups progresses generally in a non-linear fashion, so that both the nature of the data as well as the nature of the underlying processes make it necessary to employ advanced statistical methods (Sloane & Kelly, 2008). Order effects will become more pronounced as groups construct their histories and make use of them, through communication, as resources for interpreting events and planning future actions.

This paper introduces *process modeling and mining* as an approach to process analysis for CSCL. This approach occupies an interesting middle ground between particularistic models of change on one end (formalized as Markov models, for instance) and holistic models of change on the other end (represented as narratives, for instance) of the method spectrum. We begin by characterizing the position of Process Modeling in the overall landscape of process analysis methods and then demonstrate how it can be used to analyze temporal aspects of prototypical CSCL/CSCW data: group decision making that takes place in a chat. The discussion and conclusion describe the models that were produced, and discuss the areas for future research.

Process analysis

Synthesizing earlier reviews on process analysis methods in HCI, CSCW, and CSCL such as Sanderson & Fisher (1994), Olson, Herbsleb & Rueter, (1994) and Ritter & Larkin (1994) and incorporating related work in organizational science (in particular Poole, van de Ven, Dooley, & Holmes, 2000), we distinguish between *atomistic* and *holistic* views of process. The main rationale for this distinction is a view of a process either as being made up of *particulars,* the ordering of which is governed by an underlying law-like process, or a view of process as a whole, a plot-like structure. Along this *granularity* dimension, we can distinguish between (time) series analysis, (event) sequence analysis, and narrative methods.

A second important distinction concerns the *unit of analysis*, which can be variables or events. Variables are attributes of fixed entities defined by measurement (e.g., with a scale) or by a coding and counting procedure. The level of motivation to continue with group work is an example for a variable (typically measured with a Likert scale), the frequency of altruistic behavior displayed in a group is another, typically assessed by *coding and counting*, i.e. content analysis (Strijbos, Martens, Prins, & Jochems, 2006; Wever, Schellens, Valcke, & Keer, 2006). What counts as an event is basically up to the researcher, constrained by theory and informed by research goals; events are not 'raw data', not incidents. Combining these two dimensions of Granularity and Unit of Analysis yields a classification of exemplary process analysis methods as depicted in Table 1 (for more details see Reimann, 2007).

Table 1: Examples for methods classified according to Granularity and Unit of Analysis

Unit of Analysis	**Atomistic**	**← Granularity of process →**	**Holistic**
	Series	**Sequence**	**Narrative**
Variable-oriented	Time series analysis	Quantitative parameters of sequences (e.g. length)	Quantitative parameters of narratives (e.g. word frequencies)
Event-oriented	Stochastic modelling, e.g. Markov models.	Optimal matching methods; Graphical methods; Process-modelling and – mining.	Ethnomethodological approaches; conversation analysis.

Our focus in this paper is on a view of process as a *sequence of events*. 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. Again, we are not suggesting a strong distinction here, only a heuristically useful one.

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. Staying with sequences from now on, the question arises 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. Abbott & Hrycak, 1990) or to cluster observed sequences in other ways (Kaufman & Rousseeuw, 1990). Another approach for pattern identification is to rely on graphical representations and use visual cues to group sequences into clusters (e.g., Suthers, 2006). Furthermore, representations of processes in a graphical notation format can be quantitatively analyzed, see for instance (Winne & Nesbit, 1995).

A different way to look at sequences is to see them as *generated* by an abstract process - to see 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 (Weinberger & Fischer, 2006) or from project-based cooperation. In such cases, we can think of groups as *activity systems*—as entities that carry out their projects (Engeström, 1999; McGrath & Tschan, 2004), and of a log file as containing at least in parts records of these structured (planned, coordinated) activities.

In the rest of this paper, we will concentrate on this stance--*seeing observed event sequences as instances of (one or more, but few) process models*--because a good part of CSCL research addresses situations where groups act as activity systems, but where the methodological consequences of this on the analysis of data are as of yet not fully taken into account. We advocate further considering in CSCL research *discreet event model* (DEM) formalisms more intensively, because in this model class synchronicity of events, or parallelism, can be represented and analysed. This is particularly relevant in situations where tasks are accomplished based on a division of labour, as is often the case for project-based teams and with those forms of scripted collaboration that include parallel lines of activities.

Process Modeling as a Method for Sequence Analysis

Process modeling has roots in Business IT and theoretical computer science rather than research computing. 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. Processes can be modeled in many forms, e.g. using a system dynamics formalism for continuous process models (Sterman, 2000). The class of process models we want to concentrate on here pertain to the large class of discrete event systems (Cassandras, 1993). Finite state machines are one type of modeling language that can be used to describe and analyze discrete events systems. Another one is the language and theory of Petri nets (Reisig, 1985). Petri nets can be mathematically described as a bipartite directed graph with a finite set of places P, a finite set of transitions T, both represented as nodes (round and rectangular, respectively), and two sets of directed arcs, from places to transitions and from transitions to places, respectively. The Petri net shown in Figure 1, for instance, expresses the fact that all process instances start with A and end in D. It also expresses the fact that the only predecessor to B is A, the B can only be followed by D, and that possible predecessors for D are B, C, and E. Furthermore, it shows that B, C, and E can be executed in parallel, or any order. (Two "technical" transitions are included in the net, and *And Split* (AS) and an *And Join* (AJ) in order to express formally the parallelism between activities B and C).

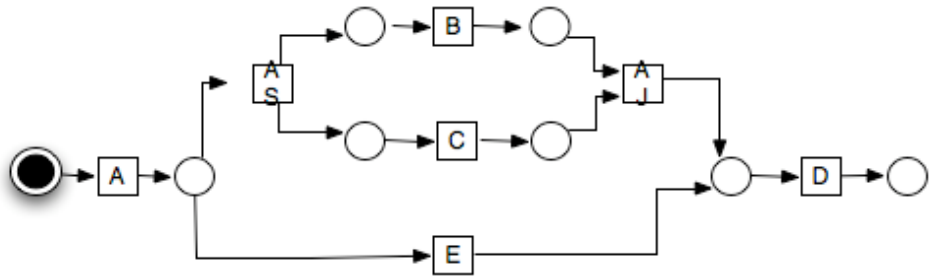


Figure 1: Example for a Petri net description of a process.

A Petri net is not only a graphical representation of a process, but Petri nets can be *executed*. That is to say, one can observe the interactions between the components and study the dynamics of the system modeled. Also, since they have formal semantics, they can be used to determine computationally if a specific activity sequence is commensurate with a model or not; like a grammar, a model can 'parse' an activity sequence. For the same reason, one can use them to simulate potential (non-observed) model behavior computationally, and to compare different models with respect to certain formal parameters. The graphical notation can be exploited for learning purposes; for instance the graphical representations could be made an object for comparison and reflection for the group members, i.e. serve as a mirroring or feedback device (Kay, Yacef, & Reimann, 2007).

In terms of the terminology introduced in this paper, DEMs, e.g., expressed in a Petri net notation, constitute a holistic view of a process: a process has a beginning and an end, it comprises events (activities), and the possible event/activity sequences are subject to more or less numerous constraints. Even a simple Petri net is a basic, but powerful language to represent for instance the logic of a group script. While Petri nets are one out of many possible formalisms to express a process succinctly, they have another advantage: they can be automatically discovered from performance data by process mining, a variant of data mining.

A specific class of data mining methods can be applied in situations where we can expect that a group realizes a multi-step process over time. This would be the case, for instance, when the group behavior is controlled by a script (Dillenbourg & Hong, 2008; Kollar, Fischer, & Hesse, 2006), or when the nature of the task suggests a specific sequence of activities, such as phases of a decision making process (Poole & Roth, 1989b). Process model mining (or process mining, for short) assumes that (a subset of) observed activities can be related to one or more processes, or in other words that (a subset of) observed activities constitutes an instance of a process. We look next at such a case: a normative model of group decision making is seen as constituting a process, the enacted decision sequences as instances thereof.

The Temporal Nature of Decision Making in Groups

Decision making is an important element of all forms or team work where the task is not completely routine. It is also an element of groups that have learning as their main purpose, to the extent that their interaction and communication is not completely prescribed. One of the first publications regarding group decision making appeared in the 1950s, when Bales and Strodtbeck (1951) developed a method for coding interaction that occurred in small group meetings. They hypothesized that members of problem solving groups tend to go through several distinct phases, emphasizing problems of *orientation* at first (e.g. giving or asking for information and clarifications), then problems of *evaluation* (e.g. giving or asking for opinions and evaluations) and finally problems of *control* (e.g. suggesting or asking for directions or ways of action).

In the 1980s, (Poole & Roth, 1989b) challenged the unitary phase models that Bales and Strodtbeck introduced in 1951. They argued that decision behaviour in groups does not follow a linear set of phases in order to reach a decision, but is a much more complex process instead. They proposed a contingency model, stating that multiple variables like task nature or group composition cause differences in the group's developmental path. 'Rather than picturing group decision making as a series of phasic "blocks" dropped one after another into sequence, the model describes development as a series of intertwining threads of activity that evolve simultaneously and interlock in different patterns over time' (p. 328).

To answer the question of *how* these variables cause variations in decision making patterns, it was essential for Poole and Roth to build a typology of the existing decision paths. In order to do that, they studied 47 decisions made by 29 groups. Their research procedure consisted of four steps. First, they coded decision making interaction with the Decision Function Coding Scheme (DFCS), which categorizes statements based on their function in the group. Basic categories of functions are: problem analysis, group orientation, and solution activities. Second, they grouped coherent statements together and identified decision phases. Third, to get an overview, they plotted this sequence of phases on a timeline. Fourth, they applied statistical methods to group these timelines and thus developed a typology of different patterns.

Their results showed that 11 out of the 47 decisions had a unitary sequence of activities, 22 had a complex cyclic structure in which groups cycled through problem-solution sequences multiple times, and the remaining 14 decisions followed a solution-oriented path, in which solution development dominated over problem statements and group orientation. This indicates that group decision making is indeed more complex than the unitary phase models assumed. In follow-up research, Poole and Roth tested the contingency model to find out which variables predicted the group's decision path (Poole & Roth, 1989a). They examined how differences in three groups of independent variables, being objective task characteristics (e.g. openness, goal clarity), group task characteristics (e.g. novelty, innovativeness), and group structural characteristics (e.g. cohesiveness, size), affected the decision making process. The analysis showed that group structure and task characteristics were the most powerful predictors of the unitariness and solution orientation of a group's decision path.

Since its emergence in the 1980s, the overall goal of functional research has been to gain understanding of how these communication functions relate to the *effectiveness* of the group decision (Gouran, 1999; Poole, 1999). One of the most sophisticated functional theories is Gouran & Hirokawa's Functional Theory of Group Decision Making, which holds that communication has to fulfil several distinct task requirements in order to result in effective decision making (Gouran & Hirokawa, 1996). More specifically, effective decision making depends on five factors: 1) the group's understanding of the issue at hand, including the nature and possible causes of the problem, 2) the group's understanding of the criteria for an acceptable solution alternative, 3) generation of as many realistic and applicable solution alternatives as possible, 4) evaluation of the positive properties or consequences of the generated alternatives, and 5) evaluation of the negative properties or consequences of the generated alternatives. Groups that communicate in a way that fulfils all these requirements will make better decisions than groups that do not.

Over the past two decades, research on Gouran & Hirokawa's Functional Theory has build up a solid base of evidence and was expanded to incorporate environmental factors (cognitive, affiliative and egocentric constraints) that affect *how well* communication fulfils the task requirements (Gouran & Hirokawa, 1996; Wittenbaum et al., 2004). However, more detailed research into the relation between specific communication functions and group performance was inconclusive. In earlier work, researchers found a correlation between the evaluation of *positive* aspects of solution alternatives and the group's performance (Graham, Papa, & McPherson, 1997; Propp & Nelson, 1996), but in a more recent meta-analysis, Orlitzky and Hirokawa (2001) found that the assessment of *negative* consequences of alternatives had the strongest correlation to group effectiveness. According to them, the nature of the task most likely dictates whether it is more important to assess positive or negative aspects of solutions (Orlitzky & Hirokawa, 2001).

On the other hand, problem analysis and development of solution criteria are two communication functions that were found to affect decision performance in almost all cases (Graham et al., 1997; Hirokawa & Salazar, 1999). Interestingly, the generation of alternatives seems to be largely *unrelated* to decision making performance. Brainstorming and idea-generation are often highly valued, but these results indicate that no clear relationship exists between the amount of ideas and the quality of the solution. One explanation for this observation is that groups that spend a lot of time on idea-generation have less time left to perform functions that are more important to decision quality (Orlitzky & Hirokawa, 2001).

This Study

Our research question pertains to the development of group decisions over time. We specifically build on former work (Poole & Roth, 1989a, 1989b) that introduces the notion that group decision making neither follows a unitary sequence (e.g. orientation – evaluation – control) nor is it completely contingent on characteristics of the situation, but that groups actively structure their decisions. In this conceptualisation, normative decision models

play the role or a resource that groups can and will access and employ, but that do not completely account for groups' behaviour. Poole's model assumes further that groups work on multiple "threads" at the same time, and decisions with respect to all threads are mingled together in observable behaviour. These threads are: task process behaviour (e.g. orientation, evaluation), relationship management (e.g., conflict, integration), and topical focus (substantive issues involved in the task).

Setup

Data were obtained using a tool called Snooker (Ullman, Peters, & Reimann, 2005) from a group of graduate students who worked on a complex problem involving a system dynamics task without meeting face to face over a number of sessions (Reimann, Thompson, & Weinel, 2007). Weekly chat meetings were conducted including all students and a session moderator, but participants were free to meet in the chat at other times. Students were given instructions regarding management of their teams. Part of this was that the lecturer and tutor would not be micro-managing the team. Instead, students would be expected to coordinate their own work within their team. This required frequent decision making regarding not only aspects of the task (e.g. which elements to include in the solution) but also regarding aspects of group work such as distribution of tasks and coordination of on-line meetings.

The participants in this study were two groups of postgraduate students who enrolled for an on-line course on system dynamics. Group A consisted of three female students and one male student. These students were scaffolded in managing their group processes, however received little scaffolding on the requirements of the task. Group B consisted of fewer students, three female in total, however one student missed a large component of the course, and so only two of the group members participated in the first three chat sessions only involved two of the three group members. Group B received much more guidance than Group A in the requirements of the task. They did have some experience in managing the group processes using the available tools (both the wiki and Snooker), however it was less extensive than that experienced by students in Group A.

The learning environment combined synchronous and asynchronous communication components. The main asynchronous collaboration medium used in this course was a wiki engine, not further discussed here. Weekly chat meetings were conducted including all students and a session moderator (one of the lecturers). The chat environment Snooker was accessible through any web browser, and combines a chat area, a notes area, and a shared whiteboard.

Coding

To be able to reliably describe processes, content analysis of communication transcripts should involve segmenting the data stream into meaningful units, and coding these units with a theoretical coding scheme (Poole, Ven, Dooley, & Holmes, 2000). Each chat statement was considered one meaningful unit. Minor unitizing adjustments were made to the dataset to simplify the coding process. For example, spelling or typing corrections and statements that consisted of a single question mark were excluded. In addition, when a statement consisted of more than one sentence and both sentences could be assigned a different code, the statement was be split, and the second sentence was set to follow the first with a one−second delay. These modifications were infrequent, however, and for the most part statements were left untouched. Since there were very few changes made to the segments, there was no need for a segmentation reliability measure (Strijbos, Martens, Prins, & Jochems, 2006). The Decision Function Coding Scheme (DFCS, Poole & Holmes, 1995; Poole & Roth, 1989a) was used to code each of these statements in the chat log. The DFCS is a well−established coding scheme, used to categorize statements according to the function that they serve in group communication. It makes use of six categories: 1) problem definition, 2) orientation, 3) solution development, 4) non−task, 5) simple agreement, and 6) simple disagreement. Table 1 shows an overview these categories. Small modifications were made to the original coding scheme by Poole and Holmes (1995): The categories were simplified and phasic markers were omitted to better suit the scheme to the current research questions.

Problem definition statements are statements that relate to the group's understanding of the problem or decision making task at hand. For example: "We need to decide if we should use a more recent timeframe". Orientation statements are statements that relate to the group process. These statements are attempts to orient the group, providing information and steering the group in a direction, or reflecting on the group process: "Why don't we do what we suggested from the start?" or "Anyone got any suggestions?". Solution development statements are all statements that are related to solutions. Statements could set criteria for solutions, suggest alternatives, elaborate on an alternative, evaluate an alternative, or confirm the solution (e.g. "Can I just confirm that we have all agreed to go with Charles' proposal?"). The non-task statements are statements that are not aimed toward making a decision or solving the problem. These include greetings, the making of appointments and other coordination statements. Finally, the simple agreement/disagreement statements are all statements that consist of a simple "Yes" or "No", or indicate agreement or disagreement in any other way. This coding scheme was applied to utterances that were classified as being part of a larger decision making process, of which 35 instances were identified in the chat log of the two groups.

A first coder analysed the complete chat log for both groups. The second coder coded 9 of the 35 decision instances (25%) and agreement on these instances yielded a Cohen's Kappa of .65. After a final discussion session, differences were reviewed and codes were changed, which resulted in a final agreement of .98.

Table 2: The Decision Function Coding Scheme

1	Problem definition: Statements that define or state the causes behind a problem, or evaluate problem analysis statements
2	Orientation: Statements that attempt to orient or guide the group's processes, including simple repetitions of others' statements or reflections on the group process
3	Solution development
3a.	Solution analysis: Statements that concern new criteria for decision making or general parameters for solutions
3b.	Solution suggestion: Suggestions of alternatives
3c.	Solution elaboration: Statements that provide detail or elaborate on a previously stated alternative. They are neutral in character and provide ideas or information about alternatives
3d.	Solution evaluation: Statements that evaluate alternatives and give reasons, implicit or explicit, for these evaluations (+ for positive, - for negative valence)
3e.	Solution confirmation: Statements that state the decision in its final form or ask the group for a final confirmation.
4	Non-task: Statements that do not have anything to do with the decision task
5	Simple agreement
6	Simple disagreement

Process Mining Algorithm and Tool

Given the expectation that the decision development would not be a simple unitary sequence, and that the chat data would contain "noise" from a theoretical point of view even after data cleaning, a non-deterministic method for process modelling was needed. While the above Petri Net model class has many advantages, this model class is not easily fitted to data that contain noise (i.e., not all events can be seen as belonging to the model) and/or are incomplete (not all model elements of the model are observed at least one time). What is needed for noisy and/or incomplete data is a model type that makes less strong assumptions on the relation between events observed and relations in the model. One such model class are *dependency graphs*, along with a *HeuristicsmMiner algorithm* to discover models from event logs (Weijters, Aalst, & Medeiros, 2006).

The HeuristicsMiner uses a frequency based metric to express the degree of certainty of a dependency relation between two events *A* and *B* based on an event log *W*, expressed as: $A \Rightarrow_w B$. With $|a >_w b|$ standing for the *number* of times *a* is followed by *b* ($a >_w b$), the metric is calculated as:

$$A \Rightarrow_W B = \left(\frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$$

In words: The number of times a is followed by b is subtracted from number of times a follows b and this difference is divided by the sum of these two relations, plus 1. This metric takes values between 1.0 and -1.0, with a value close to 1.0 indicating a high certainty that b follows a, and values close to -1.0 an almost definite certainty of the reverse (a follows b). The metric's value is dependent on the number of cases. For instance, if b follows a 5 out of 6 times, and the other order never occurs, then a =>w b = 5/6 = 0.833. If a follows a 50 times and the other order never occurs, then the value is 50/51 = .980. Instead of using a fixed value for $a \Rightarrow_w b$ as the threshold, the heuristic to take the highest score to decide which relation to put into the dependency graph is appropriate if we request that all observed activities should be connected. The HeuristicsMiner algorithm can not only deal with noisy and incomplete event logs, but also with short loops (e.g. ACCB, ACCCB) and with non-free-choice situations: in some process models the choice between two activities depends on choices made in other parts of the process model.

To perform the process mining procedure, the coded transcript needed to be imported into the ProM tool (Aalst van der et al., 2007). Since event logs exist in many different file formats, the ProM tool works with one generic XML format, Mining−XML or MXML for short. For the current study, a special plug−in was developed to convert the Snooker chat transcript, which was stored in a Microsoft Excel file, into MXML format. This MXML file was then imported into ProM and the data was analyzed. Before running the analysis, a filter was set up to discard any events of the non−task statement category. Even though unrelated chat statements occur regularly, it is improbable that they make up a specific phase in the decision process, and are therefore discarded from the model generating process.

Results

In Group A, the final event log consisted of 1115 events. These 1115 events were spread out over 23 decision instances. The mean number of events per instance was 48, with a minimum of six events per instance and a maximum of 234 events per instance. To clarify, this means that the group reached their final decision in six statements in one case, but the longest decision took 234 statements. On average, to come to a decision took 48 contributions. Group B produced a final transcript of 324 events, spread out over 12 decision instances. The average number of statements that were necessary to reach a solution was 27, with a minimum of five and a maximum of 59. Table 3 shows the numbers and frequencies of all the decision functions. In both groups, the majority of statements related to orientation of the group, indicating an important role for monitoring and guiding group processes. Concerning the solution related statements; the generation of solution alternatives appears to be the most frequent activity, followed by solution elaboration. Confirming solutions and generating criteria for solutions were performed less frequently, and evaluating solutions was the least frequent activity. Interestingly, negative responses as indicated by disagreement and negative evaluations were very rare, making up just about one per cent of all their disagreement. Another interesting finding is the low frequency of problem definition statements. Only in 5.3% of all the statements did the students refer to the actual problem or decision at hand. Finally, there is a considerable amount of non−task communication, indicating that there were off−topic conversations even when the group was engaged in a decision case.

Table 3: Frequencies of decision functions

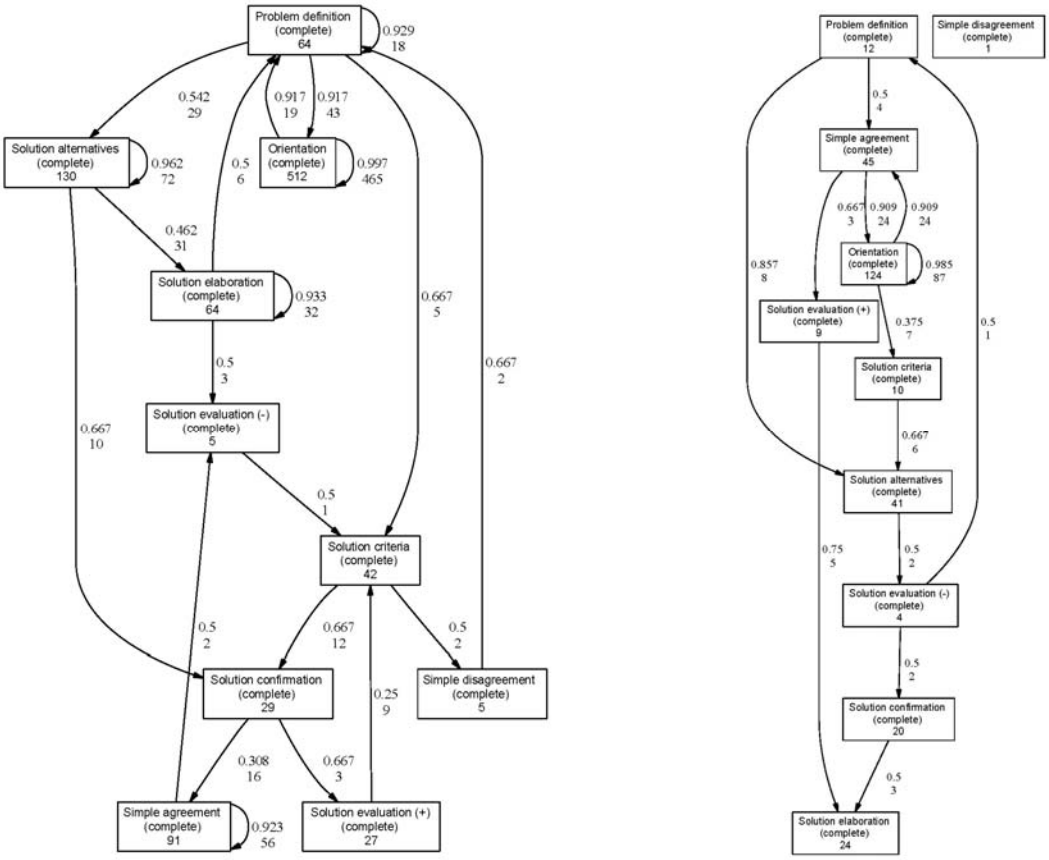
	Group A		Group B		Total	
Function	*N*	Frequency	*N*	Frequency	*N*	Frequency
Problem definition	64	5.7%	12	3.7%	76	5.3%
Orientation	512	45.9%	124	38.3%	636	44.2%
Solution criteria	42	3.8%	10	3.1%	52	3.6%
Solution alternatives	130	11.7%	41	12.7%	171	11.9%
Solution elaboration	64	5.7%	24	7.4%	88	6.1%
Solution evaluation (positive)	27	2.4%	9	2.8%	36	2.5%
Solution evaluation (negative)	5	0.4%	4	1.2%	9	0.6%
Solution confirmation	29	2.6%	20	6.2%	49	3.4%
Non-task	146	13.1%	34	10.5%	180	12.5%
Simple agreement	91	8.2%	45	13.9%	136	9.5%
Simple disagreement	5	0.4%	1	0.3%	6	0.4%

Note: frequencies are rounded to one decimal, causing minor rounding errors.

In a second step, we examined the *development* of decisions, using proces mining techniques. The HeuristicsMiner's threshold parameters were kept at their default values of 0.9 (dependency threshold), 10 (positive observations threshold), and 0.05 (relative−to−best threshold). These thresholds indicate which event dependencies are added to the model in addition to the best dependencies. First, all dependencies with a value higher than the dependency threshold are added. Second, regardless of the dependency value, each relation that occurs at least as many times as the positive observation threshold is included. Finally, each dependency that differs less from the best dependency than the relative-to-best-threshold value is added as well.

The resulting dependency graphs are displayed in Figure 2 (A and B). 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.

*The Group A decision model***.** Beginning with a period of problem definition, the model in Figure 2A shows that outgoing arcs flow to discussion of solution alternatives, group orientation, and discussion of solution criteria. The dependencies scores indicate that problem definition statements are most often followed by extensive periods of group orientation. Note the high degree of cycling and the high dependencies. The group then shifts to discussion of either solution criteria or solution alternatives. In most cases, after solution alternatives are generated, the group proceeds to decision confirmation, but in some cases, generation of alternatives is followed by elaboration and evaluation. After deciding on solution criteria, the most common next event is solution confirmation. Note that in some paths, generation of alternatives does not occur. Interestingly, if there was disagreement after deciding on criteria, the group often started from the top again. The confirmation of solutions was followed by positive evaluation or agreement. In some cases the positive evaluation statements triggered discussion of the criteria again. Incidentally, after agreeing upon a confirmed solution, negative aspects of the solution were found and the group cycled back to the generation of solution criteria.



A B

Figure 2: Process models in form of transition diagrams for teams A and B

The Group B decision model. The model displayed in Figure 2B at first sight looks very linear. Also of interest is that there is an unconnected box in the model. Simple disagreement only occurred once in the complete transcript, and the miner was therefore not able to discover any relation with other events. After the problem definition, the group took one of two paths: they either moved through a phase of agreement and orientation to the generation of criteria, followed by generation alternatives, or they started generating solution alternatives directly. The latter path was more common. The solutions were then evaluated, and a final solution was confirmed. In addition, the model shows a phase of elaboration as final event in the process. There is one peculiar sequence of events where the group moves to positive evaluation of alternatives right after they agreed upon a problem definition. They then move directly to the final phase, which is solution elaboration. One possible reason for this is that the group used asynchronous communication means in the form of a wiki-platform in parallel with the Snooker tool. We observed that some of the decision events took place on the wiki platform, making them unavailable to the miner. These kinds of missing data in the event log can cause such odd sequences in the resulting model.

Conclusions

The models shown in Figure 2 do not resemble a linear, unitary phase model. They are, in accordance with previous findings (Poole & Holmes, 1995), unstructured, complex, and cyclic. The decision process takes a different path each time it is executed; looping and cycling back to previous events also occurred often. Nevertheless, the graphic representation of the models makes the differences and similarities between the groups clearly visible.

We have argued that creating *models* from interaction data has various advantages, both as a research method as well as a means to provide feedback to teams. In this paper, we had only space to illustrate their value as a process analysis method for cases where groups can be conceptualized as activity systems. In this situation, it is appropriate to see the event sequences produced by groups as at least partially generated by rules that govern groups' (accountable) work, and to formalize these rules (pertaining to division of labour, decision making etc.) as a process model. The task for the researcher is then to identify the process model that accounts most parsimoniously for the observed process instances and/or to compare the fit of the observed process enactments with the stipulated group process. We see it as an advantage of this approach that it allows for a clean separation between observed event sequences and a model that represents these sequences in a generalized

manner. For other forms of sequence analysis, such as pattern analysis, it is more complex to decide what is an instance, and what a generalization of a set of instances. Another advantage is that specific model classes, such as the discreet event models used here, have well-understood notions of concurrency, of parallelism.

However, models are always wrong, unless they identical with the 'original'. Models are wrong because they reduce the information contained in the data modelled. Therefore, process mining using heuristics is subject to all that can go wrong with data mining (Han & Kamber, 2001) and inductive approaches in general. This means that the quality of a model depends on the quality and representativeness of the data on which it has been constructed. In addition to this general concern, process models may overfit or underfit the data. We will work with the strategy suggested by (Aalst van der et al., 2008) to reduce overfitting by "folding" regions in transition nets as displayed in Figure 2 into Petri nets. Also, we are beginning to analyze the effects of displaying process diagrams back to groups of students as a form of feedback. Research such as this will yield further insights into the value of process models both as a research tool as well as a resource for groups.

References

Aalst van der, W. M. P., Rubin, V., Verbeek, H. M. W., van Dongen, B. F., Kindler, E., & Günther, C. W. (2008). *Process mining: A two-step approach to balance between underfitting and overfitting* (No. BPM-08-01). Eindhoven, The Netherlands: Technical University of Eindhoveno. Document Number)

Aalst van der, W. M. P., van Dongen, B. F., Günther, C. W., Mans, R. S., Alves de Medeiros, A. K., Rozinat, A., et al. (2007). PROM4.0: Comprehensive support for real process analysis. In J. Kleijn & A. Yakovlev (Eds.), *Petri nets and other models of concurrency (Proceedings 28th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2007, Siedcle, Poland, June 25-29, 2007)* (pp. 484-494). Berlin: Springer.

Abbott, A., & Hrycak, A. (1990). Measuring resemblance in sequence data: an optimal matching analysis of musicians' careers. *American Journal of Sociology, 96*, 144-185.

Arrow, H., McGrath, J. E., & Behrdal, J. (2000). *Small Groups as Complex Systems: Formation, Co-ordination, Development and Adaptation*. Thousand Oaks, California: Sage Publications.

Bales, R. F., & Strodtbeck, F. L. (1951). Phases in Group Problem Solving. *Journal of Abnormal and Social Psychology, 46*, 485-495.

Cassandras, C. G. (1993). *Discrete event systems*. Homewood, IL: Richard D. Irwin, Inc.

Dillenbourg, P., & Hong, F. (2008). The mechanics of CSCL macro scripts. *International Journal of Computer-supported Collaborative Learning, 3*(1), 5-24.

Engeström, Y. (1999). Activity theory and individual and social transformation. In Y. Engestroem, R. Miettinen & R.-L. Punanmäki (Eds.), *Perspectives on activity theory* (pp. 19-38). Cambridge: Cambride University Press.

Gouran, D. S. (1999). Communication in Groups: Emergence and Evolution of a Field of Study. In L. R. Frey, D. S. Gouran & M. S. Poole (Eds.), *The Handbook of Group Communication: Theory and Research* (pp. 3-36). Thousand Oaks, CA: Sage Publications.

Gouran, D. S., & Hirokawa, R. Y. (1996). Functional Theory and Communication in Decision-Making and Problem-Solving Groups: An Expanded View. In M. S. Poole & R. Y. Hirokawa (Eds.), *Communication and Group Decision Making* (2nd ed., pp. 55-80). Thousand Oaks, CA: Sage Publications.

Graham, E. E., Papa, M. J., & McPherson, M. B. (1997). An Applied Test of the Functional Communication Perspective of Small Group Decision-Making. *The Southern Communication Journal, 62*(4), 269-279.

Han, J., & Kamber, M. (2001). *Data mining: concepts and techniques*. San Francisco: Morgan Kaufman.

Hirokawa, R. Y., & Salazar, A. J. (1999). Task-Group Communication and Decision-Making Performance. In L. R. Frey, D. S. Gouran & M. S. Poole (Eds.), *The Handbook of Group Communication Theory and Practice* (pp. 167-191). Thousand Oaks, CA: Sage Publications.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data. An introduction to cluster analysis*. New York: Wiley.

Kay, J., Yacef, K., & Reimann, P. (2007). Visualisations for team learning: small teams working on long-term projects. In C. Chinn, G. Erkens & S. Puntambekar (Eds.), *Minds, mind, and society. Proceedings of the 6th International Conference on Computer-supported Collaborative Learning (CSCL 2007)* (pp. 351-353). New Brunswick, NJ: International Society of the Learning Sciences.

Kollar, I., Fischer, F., & Hesse, F. W. (2006). Collaboration scripts - a conceptual analysis. *Educational Psychological Review, 18*, 159-185.

McGrath, J. E., & Tschan, F. (2004). *Temporal matters in social psychology: Examining the role of time in the lives of groups and individuals*. Washington, DC: American Psychological Association.

Olson, G. M., Herbsleb, J. D., & Rueter, H. H. (1994). Characterizing the sequential structure of interactive behaviors through statistical and grammatical techniques. *Human-Computer Interaction, 9*(3/4), 427-472.

Orlitzky, M., & Hirokawa, R. Y. (2001). To Err is Human, To Correct for it Divine: A Meta-Analysis of Research Testing the Functional Theory of Group Decision-Making Effectiveness. *Small Group Research, 32*(3), 313-341.

Poole, M. S. (1999). Group Communication Theory. In L. R. Frey, D. S. Gouran & M. S. Poole (Eds.), *The Handbook of Group Communication: Theory and Research* (pp. 37-70). Thousand Oaks, CA: Sage Publications.

Poole, M. S., & Holmes, M. E. (1995). Decision Development in Computer-Assisted Group Decision Making. *Human Communication Research, 22*(1), 90-127.

Poole, M. S., & Roth, J. (1989a). Decision Development in Small Groups V: Test of a Contingency Model. *Human Communication Research, 15*(4), 549-589.

Poole, M. S., & Roth, J. (1989b). Decision making in small groups IV: A typology of group decision paths. *Human Communication Research, 15*(3), 323-356.

Poole, M. S., van de Ven, A., Dooley, K., & Holmes, M. E. (2000). *Organizational change and innovation processes. Theories and methods for research*. New Oxford: Oxford University Press.

Propp, K. M., & Nelson, D. (1996). Problem-Solving Performance in Naturalistic Groups: A Test of the Ecological Validity of the Functional Perspective. *Communication Studies, 47*(1-2), 35-45.

Reimann, P. (2007). Time is precious: Why process analysis is essential for CSCL (and also can help to bridge between experimental and descriptive methods). In C. Chinn, G. Erkens & S. Puntambekar (Eds.), *Minds, minds, and society. Proceedings of the Computer-supported Collaborative Learning Conference (CSCL 2007)* (pp. 598-607). New Brunswick, NJ: International Society of the Learning Sciences.

Reimann, P., Thompson, K., & Weinel, M. (2007). *Decision making patterns in virtual teams*. Paper presented at the ICCE 2007 (5-9 Nov 2007, Hiroshima, Japan).

Reisig, W. (1985). *Petri Nets. An introduction*. Berlin: Springer.

Ritter, F. E., & Larkin, J. H. (1994). Developing process models as summaries of HCI action sequences. *Human-Computer Interaction, 9*(3/4), 345-383.

Ritter, F. E., Nerb, J., Lehtinen, E., & O'Shea, T. (Eds.). (2007). *In order to learn: How the sequences of topics influences learning*. Oxford: Oxford Press.

Sanderson, P. M., & Fisher, C. (1994). Exploratory sequential data analysis: foundations. *Human-Computer Interaction, 9*(3/4), 251-317.

Schegloff, E. A. (2007). *Sequence organization in interaction*. New York: Cambridge University Press.

Schümmer, T., Strijbos, J.-W., & Berkel, T. (2005). A new direction for log file analysis in CSCL: Experiences with a spatio-temporal metric. In T. Koschmann, D. Suthers & T. W. Chan (Eds.), *Computer Supported Collaborative Learning 2005: The next 10 years!* (pp. 567-576). Mahwah, NJ: Erlbaum.

Sloane, F. C., & Kelly, A. E. (2008). Design research and the study of change. In A. E. Kelly, R. A. Lesh & J. Y. Baek (Eds.), *Handbook of design research methods in education* (pp. 441-448). New York: Routledge.

Stahl, G. (2006). *Group cognition: computer support for building collaborative knowledge*: MIT Press.

Sterman, J. D. (2000). *Business Dynamics. Systems Thinking and Modeling for a Complex World*. New York: McGraw-Hill.

Strijbos, J. W., Martens, R. L., Prins, F. J., & Jochems, W. M. G. (2006). Content analysis: What are they talking about? *Computers & Education, 46*, 29-48.

Suthers, D. D. (2006). A qualitative analysis of collaborative knowledge construction through shared representations. *Research and Practice in Technology Enhanced Learning, 1*(2), 115-142.

Ullman, A., Peters, D., & Reimann, P. (2005). *Developing a research supportive web-based learning System.* Paper presented at the ODLAA Conference 2005, Adelaide.

Weijters, A. J. M. M., Aalst, W. M. P. v. d., & Medeiros, A. K. A. d. (2006). *Process mining with the heuristics miner-algorithm. BETA Working Paper Series WP 166*. Eindhoven, NL: Eindhoven University of Technologyo. Document Number)

Weinberger, A., & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education, 46*(1), 71-95.

Wever, B. d., Schellens, T., Valcke, M., & Keer, H. v. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education, 46*, 6-28.

Winne, P. H., & Nesbit, J. C. (1995). *Graph theoretic techniques for examining patterns and strategies in students' studying: An application of LogMill*. Paper presented at the American Educational Research Association, April 1995.

Wittenbaum, G. M., Hollingshead, A. B., Paulus, P. B., Hirokawa, R. Y., Ancona, D. G., Peterson, R. S., et al. (2004). The Functional Perspective as a Lens for Understanding Groups. *Small Group Research, 35*(1), 17-43.