Identifying Collaboration Know-How from Action Patterns in Distributed Teams

Isabella Seeber, Ronald Maier

Department of Information Systems 1, University of Innsbruck, Innsbruck, Austria
{iisabella.seeber, ronald.maier}@uibk.ac.at

Abstract. Teams that are distributed through space and time face challenges in terms of coordination which endangers successful collaboration. Coordination of a team’s actions is in focus of this paper since effective coordination is one predictor of team performance. Thus, distributed teams need knowledge on how to effectively coordinate their activities. Research in the domain of collaboration engineering suggests the concept of thinkLets, describing transferable facilitation know-how of collaboration, as a means to support coordination efforts in teams. This position paper presents a conceptual model on how process mining can be used to extract collaboration know-how from user actions captured in distributed team settings. By relating mined user actions with thinkLets, expressed as rule-based action patterns, we can further extend our understanding of team coordination by deducing in which phase a team is currently in or to what extent the team holds collaboration know-how. To exemplify the usage of the conceptual model, communication logs from an experiment demonstrate how action patterns can be identified.

Keywords: action pattern, collaboration know-how, negotiation analysis, procedural knowledge, process mining

1 Introduction

Teams are increasingly characterized by temporal and geographical dispersion. Even though collaboration technology (CT) support is evolving and rapidly changing, adequate CT support does not necessarily lead to successful collaboration [1]. Often the cause of failure lies within problems of collaboration process designs, e.g., too much time spent on non-relevant activities, or problems with goals, e.g., conflicting expectation about meeting content [2]. Collaboration research is developing rapidly, but there is still a need to get a better understanding of this phenomenon [3] since there exist many factors that influence collaboration which in turn makes it difficult to predict [4]. Recent research studies investigate the direct impact on team performance with concepts such as IT support providing contextual information [5], knowledge application [6], task interdependence [7], or coordination problems [8]. This paper puts particular focus on the last mentioned factor and related phenomena. When teams are distributed and discussions are critical, increased process management, in particu-
lar coordinating activities, was found to be a necessity compared to co-located teams to complete the task and to reach their goal [9]. In a similar observation, task organization impacted coordination problems and this in turn had effect on team performance [8]. ThinkLets, which are packaged facilitation interventions, are deemed to cope amongst others with coordination problems since these normative descriptions of how to conduct collaboration specify useful steps and structure to a team to accomplish their task and achieve their common goal successfully [10]. This paper argues that the idea of thinkLets, and here in particular the structured descriptions of collaboration activities to be performed, represent collaboration know-how, i.e. knowledge about how to communicate and integrate ideas and how to coordinate actions in teams [5].

The goal of this paper is to discuss how collaboration know-how can be identified by process mining techniques from communication logs gathered in collaborative settings. The paper proposes to translate the collaboration know-how captured in thinkLets into so-called action patterns, which are understood as machine-readable patterns of collaboration expressed by negotiation actions, sequencing information input/output as well as information on exception/condition handling. This understanding is represented in a conceptual model and parts of it are exemplified with empirical data gathered in a laboratory experiment. By comparing action patterns captured in thinkLets with actions performed by teams during run-time of collaboration as identified in communication logs, one can analyze to what extent a team applies the collaboration know-how represented in thinkLets.

Section 2 provides background on collaboration know-how and its connection to thinkLets. Section 3 presents the conceptual model and showcases what coordination actions can be deduced from thinkLets. Section 4 exemplifies application of the conceptual model, concretely the coordination of negotiation actions, based on communication log data gathered in a laboratory experiment. Section 5 concludes this position paper and gives an outlook on future work.

2 Background

This position paper brings together multiple research streams, all interlinked through the phenomenon of collaboration and investigating it from a process-oriented perspective. This section will introduce at first the concept of collaboration know-how and coordination problems in distributed teams. Then, the concept of thinkLets will be outlined which represent collaboration know-how in the form of normative descriptions how to facilitate collaboration processes to achieve predictable collaboration process and successful team performance.

Know-how as a type of knowledge relates to the understanding of how components of a system need to be put together. In other words, know-how describes knowledge about procedures [11]. Collaboration know-how describes how a team communicates and integrates its ideas but also how a team coordinates its actions and work with members of the team [5]. Thus, collaboration know-how can be applied in activities that help regulating collaboration processes, in particular communication and coordi-
nation processes. Collaboration know-how is important for a team since it has been found to be a predictor of team performance [5, 8]. A number of explanations exist why collaboration might fail, e.g., collaborators might simply be not skilled enough to communicate efficiently which leads to unsuccessful exchange of information. A lack of information exchange might possibly result in different stocks of knowledge among collaborators impacting coordination in a negative way [1]. Likewise, the structuring and organization of a task, comprising assigning roles, agreeing on goals and processes was found to impact coordination as well [8]. Teams that fail to establish these coordination mechanisms spend more time on managing their collaboration processes rather than using their time to discuss important topics of their tasks [9]. The idea of easing the transfer of collaboration know-how and here in particular when and how to apply coordination mechanisms for the overall collaboration process is a key concept in the domain of collaboration engineering (CE).

CE can be understood as an approach that combines design, facilitation and training expertise with the goal to create repeatable work practices for effective collaboration [12]. ThinkLets enclose complex intellectual capital describing a facilitation technique and, if required, a CT configuration for interactions in collaboration processes [13]. Collaboration processes are characterized by bundles of team activities of everyday practices and are recurring while the team moves towards its goal [14]. These bundles of activities can be classified into interpersonal exchange processes, relating to communication activities, task-oriented processes, relating to cooperation, coordination and information sharing activities, as well as to group-oriented processes, relating to group maintenance activities such as developing trust [15]. The CE domain differentiates between six patterns of collaboration comprising generate (move from having fewer to having more concepts), reduce (move from having many to having fewer concepts), clarify (move from having less to having more understanding), organize (move from less to more understanding of the relationships among concepts), evaluate (move from less to more understanding of the relative value of a concept), and build consensus (move from fewer to more commitment in the team) [16]. Each thinkLet is documented in a specific way, including identification (describing the name of the thinkLet), script (describing minimum instructions how to walk through a process), and a selection guideline (describing what effects the thinkLet strives to achieve). The script holds various kinds of information including specifics for process coordination, tool configuration, behavior of team members, constraints, or input requisites [16]. To exemplify this understanding the following Box. 1 represents an excerpt from the OnePage thinkLet.

<table>
<thead>
<tr>
<th>OnePage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose this thinkLet…</td>
</tr>
<tr>
<td>… to generate a few (less than 80 or so) comments on one topic</td>
</tr>
<tr>
<td>… when 5 or fewer people will brainstorm together</td>
</tr>
<tr>
<td>… when 6 or more people will brainstorm for fewer than 10 minutes</td>
</tr>
<tr>
<td>… […]</td>
</tr>
<tr>
<td>Input: the brainstorming question</td>
</tr>
<tr>
<td>Output: a set of comments in response to a brainstorming question or prompt</td>
</tr>
</tbody>
</table>
Steps:
1. Make sure the participants understand the brainstorming question or prompt.
   Say this: If you have any questions with respect to the brainstorming question
   or assignment, please speak up.
2. If necessary, facilitate a verbal discussion to address any understanding difficulties. If necessary, re-formulate the question or prompt.
3. Inform the participants of time limits, if any.
4. Let the participants contribute comments until they run out of ideas

Box 1: Excerpt from thinkLet OnePage

3 Conceptual Model and Expressing Collaboration Know-How

The conceptual model as depicted in Fig. 1 outlines our understanding of how applied collaboration know-how can be elicited from communication logs. ThinkLets represent collaboration know-how which can be expressed, to some extent, in machine-readable form as action patterns. These are stored in an action pattern repository. During run-time, CTs or any other system or software capable of generating logs of actions relevant for collaboration trace the user interaction in terms of exchanged communication as well as their use of items during collaboration. These traces are collected in action stream logs interfaced with systems such as e.g., personal desktop, databases, or any other software. In order to infer actions from communication logs, some kind of computational logic, e.g., approaches of sentiment analysis [17, 18], would need to be applied. For our purposes, we theorize that at least the following information needs to be logged: the collaborating users, actions performed, resources or items manipulated, and the corresponding timestamp. On a generic level, this work distinguishes between two types of actions comprising coordination actions, e.g., thinkLets expressed as negotiation actions, and actions that relate to manipulation. When users perform coordination actions like for example starting a brainstorming by performing the negotiation action “proposeIdea”, the log needs to capture the person initiating this statement, the action the statement was classified to, the statement itself as a comment, as well as the time the statement was explicated. In case that the action relates to a manipulation action, e.g., inserting a picture into a document, the log needs to catch in addition to the timestamp and the user, also information on the item that was affected.
The trace of manipulation actions is deemed to be important in order to identify switches between tasks. It is likely that by concentrating on negotiation actions only, mined action patterns might be inaccurate since tasks performed by knowledge workers are typically fragmented and completed with interruptions [19]. In order to identify the task a user is currently in, it has been proposed to create directed graphs whose nodes represent items, e.g., an e-mail, and whose arcs are actions, e.g., save as. A likely task is obtained by applying a sub-graph mining algorithm and by searching for frequently occurring sub-graphs [20]. Research in conjunction with task detection is still struggling with these process characteristics and it is believed that accurate task detection is still in its infancy [21]. After having detected tasks based on manipulation, the mining of action patterns could start relating coordination actions to the detected task due to the timely connection to the manipulation actions. Action patterns can then be mined from the time-ordered list holding manipulation and coordination actions. Finally, mined action patterns are compared with explicated thinkLets stored in the action pattern repository. When compensating for additional noise in data, it is assumed that the information can be elicited whether teams have collaboration know-how at their disposal or not. In principle, action patterns from multiple teams might indicate which kind of collaboration know-how is mostly applied during collaboration, but also how a particular action pattern might impact team performance measured by e.g., saving time or better quality of end product.

The aforementioned description of the conceptual model needs further operationalization since components are not yet existent, e.g., machine-readable expression of thinkLets, or not yet mature enough to reach accurate results, e.g., analyzing communication logs to infer support or opposition to a proposal and the like. Since the scope of this position paper is limited, the rest of this section will focus on how thinkLets could be translated into action patterns.

It has been outlined that thinkLets, describing prescriptions of collaboration facilitation, represent a good starting point to express collaboration know-how in machine-readable form. It is hypothesized that we can identify to what extent collaboration know-how has been applied in teams using CTs to facilitate their collaboration, especially in distributed settings. Process mining techniques and natural language pro-
cessing enable this identification. Based on the comprehensive description of thinkLets in [13], this work examines how a thinkLet could be expressed as an action pattern which can then be automatically traced in collaborative settings when supported by CTs. It should be noted that a portion of the information given in a thinkLet cannot be explicited in terms of traceable actions especially when it comes to thinkLet prompts that describe how a human facilitator should behave, e.g., “Make sure the participants understand the problem” in ThinkLet ComparativeBrainstorm. All thinkLets include prompts how a team should communicate by defining which comments or what kind of behavior is anticipated. When analyzing thinkLet descriptions to infer formalized action patterns four components for defining action patterns emerged, comprising negotiation actions, sequencing information, exception handling and statements about the input or output (see Fig 2). Negotiation actions in this paper comprise five classes of statements resulting from a negotiation model defined in [22]. A subset of these negotiation actions seems worthwhile since we do not intend to differentiate team members’ opinions on a fine granular level. This means that at e.g., the polarity of comments does not differ between support idea, argue-for idea or accept idea. Here, we adopted the negotiation actions comprising propose idea, support and challenge idea which define the polarity of opinions, as well as ask and answer question.

**Fig. 2. Components of action patterns**

**Input/Output** describes information that specifies on the one hand what kind of input needs to exist for the action pattern, e.g., “A list of topics that must be addressed by the team.”. On the other hand, output information describes the end product of the action pattern, e.g., “A set of comments organized by discussion topic”. **Exceptions/Conditions** information triggers switches between action patterns and specifies conditions. For example, the thinkLet BroomWagon of the reduce collaboration pattern triggers the end state when stating “Repeat this process [of reducing ideas] until you end up with the maximum number of issues that you want to handle from that moment onward”. **Sequencing information** defines the ordering over time of negotiation actions. Thus, they describe the process model of the thinkLet. In this regard, it is possible that thinkLets prescribe sequential occurrences of negotiation actions, e.g., occurrences of actions but also when a user should perform what e.g., “[…] you may agree […] or argue against an idea […] [13]”.

The following Box 2 describes a pseudo-code how an action pattern could be expressed to be stored in an action pattern repository. The action pattern is based on the
thinkLet OnePage that has been introduced in Box 1. This thinkLet is classified as a generate collaboration pattern and has been chosen as an example due to its rather small complexity in terms action pattern components. After specifying the necessary variables and the log format required (see line 1 through 6) the action pattern is specified. Please note that any thresholds allowing noise in data are considered here to ease comprehensibility. OnePage prescribes that only the team members can state ideas. Hence, sequencing information would only include the occurrence of proposeIdea negotiation actions (see line 10). An exception would be triggered as soon as a following entry in the log does not correspond to a “proposeIdea” negotiation action (see line 11 and 18). As soon as a proposeIdea logentry is followed by any other negotiation action, the algorithm tests whether an OnePage action pattern exists based on the conditions specified, e.g., when fewer than 5 people participate in the collaboration (see line 18 through 27).

```
1 log
2 logentry (user, action, comment, time)
3 counter = 0
4 timeoffirstproposal
5 timeoflastproposal
6 listofusers = ()
7 listofcomments = ()
8 containsOnePage (log) {
9 while (log.hasMoreElements()) {
10   entry = logentry.next()
11   if (entry.action == "proposeIdea") {
12     if (counter == 0)
13       timeoffirstproposal = entry.time
14       timeoflastproposal = entry.time
15     if (!listofusers.contains(entry.user))
16       listofusers.add(entry.user)
17       listofcomments.add(entry.comment)
18       counter++
19   } else {
20     if (counter >= 10 && counter < 80) {
21       if (listofusers.size() <= 5) return true
22     else {
23       if (timedifference(timeoflastproposal,
24         timeoffirstproposal) <= 600) return true
25     }
26   }
27   counter = 0
28   listofusers.removeAllElements()
29   listofcomments.removeAllElements()
```
Other than carefully translating thinkLets into machine-readable action patterns, the conceptual model has additional challenges. In order to understand the meaning of user generated content in terms of the proposed negotiation actions, information retrieval techniques need to be applied to reason on user generated text during run-time. There exists extensive research that deals with the automated analysis of evaluative text and that strives to infer positive or negative polarities from user generated content (see e.g., [17, 23]). This stream of research came to be known as sentiment analysis or opinion mining [24]. As to the authors’ knowledge, there does not yet exist an appropriate approach that could automatically infer the proposed negotiation actions from communication logs. The process-oriented analysis technique CoPrA [25] represents an alternative when analyzing negotiation in communication logs even though the technique cannot be applied to run-time support of distributed teams. A more detailed description of the analysis technique is provided in the subsequent section. Additionally, also an appropriate process-mining algorithm would be required to mine action patterns from user actions. The heuristic miner algorithm, which is a plugin for the process mining tool ProM exemplifies an approach fitting also for unstructured data which calculates likely relationships among actions based on action dependency information held in process logs [26].

4 Case Example

This case example illustrates how the application of collaboration know-how can be made explicit in communication logs as action patterns. Data for this case example was collected in a laboratory experiment conducted in 2010 investigating collaborative writing in distributed team settings. Communication logs of six teams were collected whereby each team consisted of three members who had the task to develop a written report proposing technical and organizational measures to improve knowledge management for a fictional company. Team members were distributed, but worked in offices on the same campus and communicated synchronously with a CT tool, which allowed synchronous writing and chatting. Subjects were students of the master program Information System or in the diploma degree Business Sciences majoring in Information Systems at the University of Innsbruck. Half of the participants were treated with high process structure, treatment groups, which included thinkLet prescriptions as well as prompts for co-authoring. The control group received low process structure, solely the task description, which asked them to perform brainstorming, outlining, drafting, and reviewing in order to develop a written report. In order to mine action patterns, the communication logs needed to include information on the
performed negotiation, the user performing it and the timestamp. Since a prototype of the proposed model does not exist, the analysis was performed manually with CoPrA [25]. CoPrA is characterized by a qualitative content analysis at the beginning and followed by a quantitative visualization of mined data. For the purpose of this paper, we applied for the qualitative content analysis a subset of the negotiation model after [22] for the qualitative content analysis. In multiple analysis cycles of the communication logs, sentences or text phrases were annotated with a corresponding negotiation action, the person who performed the action and the time when the action occurred. The following Fig. 3 depicts three teams whereby team HBQ represents a treatment group. The other teams SRR and JAP were control groups and hence visualize how negotiation occurred over time when no further collaboration facilitation was given. In this regard, the number of actions occurred are on the y-axis whereas time is represented on the x-axis in intervals of 10 minutes. Team HBQ received thinkLet prompts how to run through the patterns of generation and convergence. For the generation phase, the OnePage thinkLet was adopted asking all team members to come up with ideas without discussing them at first. During this brainstorming phase, teams were asked not to add any further comments to any of the proposals. Subsequently, teams were asked to discuss ideas and select the most relevant ones for their report.

The treatment group HBQ in Fig. 3c depicts how the team exhibits the prompted process behavior. They made many proposals, i.e. performing negotiation action “proposeIdea”, represented by the black solid line, within the first 10-20 minutes. The team started converging on their proposals after their brainstorming phase which is indicated by the dashed grey line and describes the negotiation actions “supportIdea”. It is noticeable that the team did not perform any questions or answering comments and moreover did not challenge any ideas. The control group SRR, Fig. 3b, exhibits that the negotiation action “proposeIdea” had its peak at the beginning of collaboration but does not correspond to the OnePage thinkLet since other negotiation actions occurred during the same time. Here, the team started out with posting many proposals in time interval 3. At the same time other discussion despite proposals occurred leading to the main negotiation that took place in time interval 4-5. The various types of lines indicate broad discussions such as supporting ideas represented by the dashed grey line, challenging ideas represented by the grey dotted line, asking questions related to options represented by the dashed black line and answering questions related to option represented by the solid grey line. The peculiar thing about control group JAP, Fig 3a, is that they started out proposing ideas and discussing them at the same time. However, in the time interval 5 one team member included a list of proposals into the chat which was taken from an online web resource. The team did not further discuss the proposals. Communication ceased after time interval 6 which can be explained by the fact that the team started writing the report.
Due to the small overall sample size, three treatment and three control groups took part in the experiment, no statement can be made which kind of negotiation pattern would lead to better results. However, by visualizing NAs over time, some information on collaboration know-how can be analyzed. In particular, it was shown that the treatment group actually performed the thinkLet OnePage when allowing for some noise in data.

5 Conclusion and Outlook

This paper discussed the need for collaboration know-how to support effective patterns of collaboration actions in distributed teams supported by CTs [9] since coordination problems negatively affect team performance [8]. In this regard, thinkLets can be understood to represent collaboration know-how in the form of prescriptions of actions to be performed by team members in order to facilitate effective collaboration [12]. The paper presents a conceptual model proposing to formally express thinkLets in terms of action patterns, which can be mined in communication logs. It is suggested that these action patterns consist of four components, comprising negotiation actions, sequencing information, input/output information as well as information on exception handling and conditions. These are used to describe action patterns in detail, which in turn help identifying if, and to what extent collaboration know-how has been applied in the underlying collaboration. The paper contributes towards a more formal analysis of collaboration behavior in order to improve generally the understanding of collaboration and specifically the application of collaboration know-how. By grounding the analysis in action patterns developed on the basis of thinkLets, this research also contributes towards interlinking the descriptive automatic analysis of user behavior with a specific focus on CT-mediated collaboration in distributed team settings to the prescriptive facilitation of collaboration in such teams with the help of thinkLets.

Avenues for future research are suggested for the expression of thinkLets as action patterns. In particular, work remains to be done assessing how much information of thinkLet descriptions can be automatically traced and how coding schemas would need to adapt to allow for automatized process-mining. The style of interaction within
teams such as aggressive-dominant behavior or in the other extreme passive non-participatory behavior of team members seems to be an important predictor of team performance [27]. Drawing on methods in the field of social network analysis further insights could be gathered and subsequently better interaction support provided. Moreover, thresholds need to be found in order to categorize actual behavior as action patterns described on the basis of thinkLets and whether or not the propositions of the four components of action patterns are comprehensive enough to represent the necessary collaboration know-how. Moreover, our understanding of effective collaboration behavior could be further enhanced when connecting the team’s collaboration know-how with quantifiable indicators of team performance other than less time required. This in turn paves the way towards automatically recommending action patterns as part of thinkLets during run-time of collaboration.

Acknowledgements. This work was co-funded by the European Commission under the Information and Communication Technologies theme of the 7th Framework Programme, Integrating Project ARISTOTELE (contract no. FP7-257886).

References


23. Thomas, M., Pang, B., Lee, L.: Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. Get out the vote:
Determining support or opposition from Congressional floor-debate transcripts, pp. 327-335. Association for Computational Linguistics, Sydney, Australia (2006)


