CoPrA2GO: An App for Coding Collaboration Processes

Emmanuel Nowakowski

University of Innsbruck, Innsbruck, Austria
emmanuel.nowakowski@student.uibk.ac.at

Isabella Seeber

University of Innsbruck, Innsbruck, Austria
isabella.seeber@uibk.ac.at

Ronald Maier

University of Innsbruck, Innsbruck, Austria
ronald.maier@uibk.ac.at

Fulvio Frati

Università degli Studi di Milano, Milan, Italy
fulvio.frati@unimi.it

Abstract. Organizations seek ways how to support teams in their communication but face challenges how to make communication processes measurable and visible. Past communication research came up with numerous interaction coding systems enabling the analysis of communication. However, today we see only few applications in research and practice due to the labor-intensive effort connected to coding and analysis. This paper addresses this problem and introduces the iPad application CoPrA2GO that strives to make the coding and analysis of communication more convenient and applicable for researchers and practitioners. A user acceptance test was conducted involving 4 IS graduate students coding 23 team meetings in real-time. Our findings suggest that CoPrA2GO is useful for coding communication in real-time and providing feedback immediately after meetings.

Keywords: communication analysis, CoPrA, real-time, CoPrA2GO, team performance

1 Introduction

Worldwide organizational structures have been transformed from work organized around individual jobs to team-based work structures [1]. The transformation is mainly attributed to pressures created by increasing global consolidation, innovation, and competition. There is a need for more rapid, adaptive, and flexible responses to overcome

adfa, p. 1, 2011.

these pressures [2]. Additionally, problems are more and more complex, so that no single individual has enough influence, resources, or expertise to solve the problem alone. Therefore, team collaboration has become an omnipresent feature of organizational life [3]. Given this, it is important for organizations that teams work effectively. According to Pentland [4] it is possible to predict a team's success by looking at communication data generated during collaboration. This stream of research recommends automated ways of gathering and analyzing communication by e.g., calculating the frequency of interaction exchanges [4]. Yet, it is not adequate to capture the meaning of communication. Several interaction analysis approaches [e.g., 5, 7, 8, and 9] have been suggested to gather and analyze communication and its content. All of these approaches require the manual creation of communication logs to deduce the meaning of communication [7, 8, and 9]. Interaction coding systems such as IPA [8], SYMLOG [16], DFCS [9], and TEMPO [15] are usually independent of software tools and consequently additional information in relation to the communication act such as timestamp or name of the team member needs to be included by hand. For further analysis additional extra human manipulation is required to transfer communication logs into tools for analysis to calculate, e.g., depth or breadth of discussions and participation. Consequently, researchers and practitioners are faced with labor-intensive and time-consuming efforts to analyze team communication [5]. Better methods are needed that allow analyzing team behavior and team performance in a discreet, flexible and real-time manner [10, 11].

This paper strives to contribute to this call for better methods and presents a design artifact, which enables IT-supported coding of communication. This design artifact is implemented as an iPad application, named CoPrA2GO, which adopts a range of communication acts as defined in the COllaboration PRocess Analysis (CoPrA) technique [5]. The app allows for real-time coding of communication and is interfaced with the CoPrA Tool [12] for process analysis. We evaluated the usefulness of the tool by conducting a laboratory experiment in which four IS graduate students coded communication of 23 teams working on a decision-making task in real-time. The coders were interviewed in a subsequent focus group interview to gain insights into their experience with using the tool. The following three research questions were stated to assess the acceptability of the tool: *(1) For which purpose is CoPrA2GO used?*, *(2) What makes CoPrA2GO useable?,* and *(3) In which settings could CoPrA2GO be used?*.

The paper is structured as follows. Section 2 provides the background on team effectiveness and communication coding. Section 3 introduces the CoPrA2GO functionality. Section 4 describes the laboratory experiment and the focus group interview in which the application was tested for user acceptance. Section 5 reports the results of the CoPrA2GO user acceptance test. Finally, Section 6 and 7 contain our discussion and conclusion.

2 Background

Teams gain more and more importance and so does measuring team effectiveness [2, 3]. According to Cohen and Bailey [13] effectiveness can be measured along three dimensions comprising performance outcomes (e.g. productivity, response times, innovation), behavioral outcomes (e.g. turnover, absence, safety), and member attitudes (e.g. commitment, employee satisfaction, trust in management). Team performance is context specific, which makes it difficult to define criteria that are generalizable to other teams, the organization or even beyond [6]. In this paper, team effectiveness is primarily based on team performance composites, which are shaped by team behavior patterns evolving through social interaction. Firstly, because it is considered as important in organizational behavior and human resource management literature [14]. Secondly, because it is observable with the help of automated tools, and thirdly, because we need better measurement of factors influencing team effectiveness [11].

For the automated analysis of team performance based on communication, Pentland [4] proposed a composite of three performance criteria that he refers to as energy, engagement and exploration. Where energy denotes the individual participation in the team, engagement the communication of team members within the team, and exploration the communication of team members with other teams. Additionally, Pentland [4] states that the most important factor for high performing teams is the balance between energy and engagement.

There are long-established approaches for the analysis of team behavior based on communication. Examples are the Interaction Protocol Analysis (IPA) [8], the Decision Functioning Coding System (DFCS) [9], the TEMPO system [15], and the SYMLOG methodology [16]. For these traditional approaches, the researcher first needs to transcribe video and/or audiotaped communication, perform a coding procedure on the transcripts, and analyze the communication logs that resulted from the previous step. Each code describes the meaning of the underlying information [17]. The whole procedure aims at reducing the complexity of the team's communication to a simpler set of categories [18]. Each of the above mentioned approaches differ to a certain extent to the kind of team behavior they can deduce. For example, the IPA framework [8] includes categories for coordination and emotions and is, like the DFCS [9], capable of coding task-related communication. SYMLOG [16] and TEMPO [15] are further developments of IPA [8] and allow to create team member profiles and distributions of team behavior (for taskwork and teamwork), respectively. A further advancement in communication analysis is represented by the CoPrA technique. The technique differs from other interaction analysis methods in such sense that it puts emphasis on aggregating communication to topics during its data preparation phase [19].

3 Artifact Description: CoPrA2Go App

The basis of our work builds upon the aforementioned CoPrA technique. The CoPrA2GO application supports two major activities of collaboration analysis, comprising real-time coding of communication (data preparation) and mining patterns of team behavior (data analysis). Its main functionality is to facilitate real-time coding of communication. CoPrA2GO adopts the coding schema of the CoPrA technique to ensure compatibility with the CoPrA Tool. CoPrA2GO consists of five screens, in Objective-C so-called view controllers, which guide the coder from the login screen to the coding screen. During coding, the app generates an MXML file in the background, which allows to be analyzed by the CoPrA Tool for mining team behavior patterns and whose results are returned to the iPad screen (see Fig. 1). Please refer to [12] for more information on the mining of team behavior patterns. The remainder of the section describes the intended use of CoPrA2GO and its interface to the CoPrA Tool.

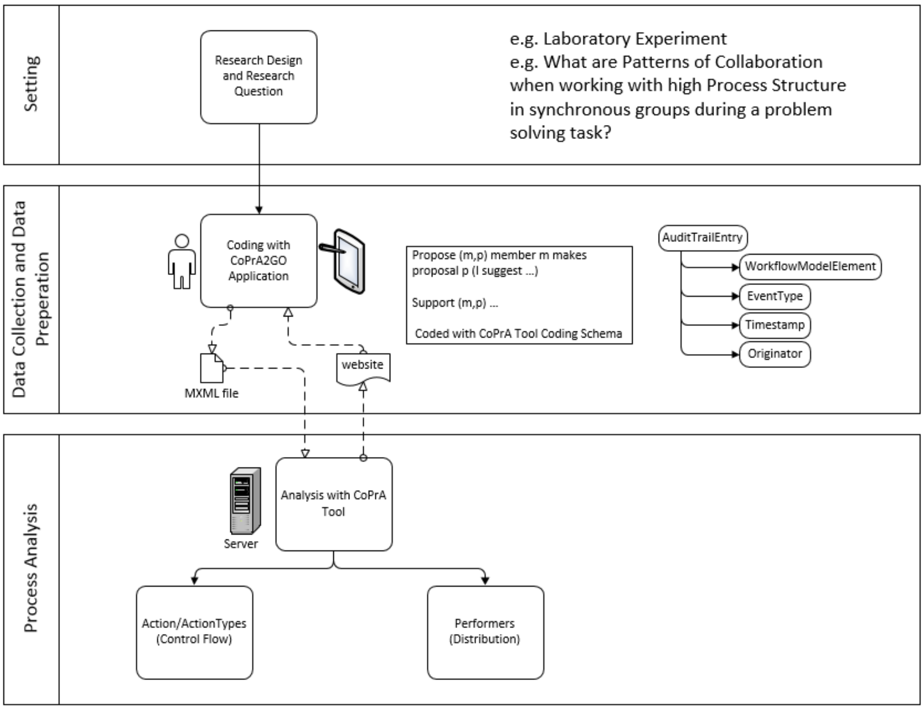

Fig. 1. CoPrA2GO for Real-Time Analysis.

Pre-Conditions

A pre-condition of using CoPrA2GO is that any coder is familiar with the communication actions that can be used for coding communication. The code book contains 18 codes including codes such as propose idea, ask for clarification, or support idea (see Appendix for a more detailed description). In addition, all coders need to be familiar with the app interface so that the respective codes can be quickly found. This ensures the on-the-fly interpretation of communication is possible and does not hinder the use of CoPrA2GO by longer thinking about code meanings and location.

Observer Registration View Controller

The design goal is that the user engagement with the application is aided by a visual representation of a meeting that looks like a physical meeting room. Therefore, the first view controller looks like a doorplate and is the entrance to the room. The coder has to register by entering their name (or a nickname) in the text field, to make it traceable who coded the specific collaboration session. After the registration is finished, the coder is able to continue to the next view controller by hitting the submit button.

Choose Table Design View Controller

In this view the coder has to choose one of four table designs. Together with the next step, group setup view controller, the coder recreates a virtual environment that reflects the physical setting including team members, facilitator, and meeting table. This is considered to ease the cognitive load for coding communication as the drag-and-drop of communication acts onto team members or the facilitator on the iPad screen is similar to their location in the physical environment.

Group Setup View Controller

After that, the coder is forwarded to the group setup view controller, used for the MXML file generation. Here, the coder enters the names of the facilitator and the team members for later placement in the virtual meeting room. Furthermore, the coder includes the server address of the CoPrA Tool server for sending the MXML for further analysis. The number of team members is currently limited to six plus one facilitator. The number of team members could be easily adapted by adjusting the underlying array, table view, and restriction. The data collected here is saved and passed to the observation view controller.

Observation Screen View Controller

Fig. 2 depicts the observation screen in which the communication coding takes place. We first describe the elements seen on the screen identified by 1 to 7 in Fig. 2 and then the actions that can be coded. In particular, (1) shows the text field where the Task ID has to be entered, (2) depicts the member labels, initially placed on the left and right hand side of the screen, which have the names specified in the group setup view controller, (3) shows the play/pause mechanism for the timer functionality, (4) depicts all the 18 code buttons from the CoPrA code book, (5) shows the activity stream that contains the last seven coded actions, (6) points out the *undo* and *redo* mechanism for the activity stream, which has also effects on the resulting MXML file, and finally, (7) shows the facilitator, who is guiding the team members through the collaboration session.

To be able to start the coding procedure, the coder first has to enter the Task ID to make communication acts assignable to a specific task. The second step is to drag-and-drop the representations of the team members on their actual sitting position at the table. After these steps are fulfilled, the play button for the timer, at the bottom on the left side, has to be hit to enable the assignment of timestamps to any coded communication action. Once the timer runs, the actual coding activity can begin.

Each code button (4) can be moved per drag-and-drop and is able to detect collisions with the member labels. Therefore, if for example the provision button collides with the label of *member1* (2), it is logged that *member1* performed, e.g., a provision action at a specific time. Furthermore, the facilitator (7), displayed as the figure behind the desk, has also a collision detection. After each code assignment, the activity stream (5) adds the latest code assignment at the top of the list and moves the older one field below. Additionally, with the undo and redo buttons (6) on the right side, it is possible to delete the last performed actions, if an error occurred, or to restore the last deleted action. Furthermore, the Task ID (1) can be changed during the coding procedure to ensure that after a new task has started, the corresponding collaboration codes are assigned to the correct task, e.g., evaluation or idea. That has the effect of generating a new "ProcessInstance" in the MXML file. The definition of task and the corresponding Task ID has to happen before the actual use of the application.

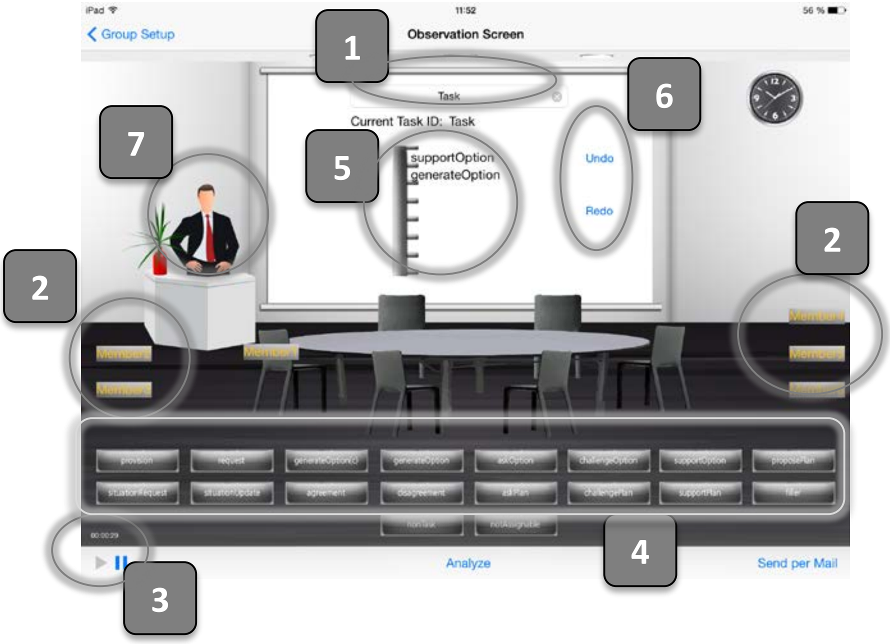

Fig. 2. Observation Screen View Controller.

After the collaboration session is finished, the pause button (3) has to be hit, in order to activate the "Analyze" and "Send per Mail" buttons. This is implemented to prevent errors, which could happen by accidentally tapping the "Analyze" button during the coding procedure.

At the end of the coding, the coder can choose between sending the generated MXML file per email to further analyze it by using, e.g., ProM [20], or to send it directly to the CoPrA Tool server for further analysis.

Web View Controller

This controller will be activated when the analyze option in the observation screen view controller is chosen. At this point, the connection to the CoPrA Tool is established and the MXML file is sent to the server for further analyses. In this controller, the coder can choose which analysis methods to perform on the MXML files sent to the server.

In fact, the CoPrA Tool applies a set of predefined metrics to the communication logs and allows the visualization of results, both in distribution and flow perspectives. The results are then shown in the CoPrA2GO app in form of tables or graphs.

4 Method

This paper describes a design science research study reporting on the activities of the research cycles comprising rigor cycle, design cycle, and relevance cycle [21]. In May 2014, the CoPrA2GO tool was tested in the context of a laboratory experiment at the University of Innsbruck collecting data from 92 undergraduate students that were randomly assigned to 23 teams. The collaboration task is named Norvos and represents a decision making task. The task was adapted from an existing one and developed further to better fit a student context. It is about a flooding that hit the city of Norvos. The goal of collaboration is to decide on supporting measures to deal with the aftermath of the flooding (e.g. organizing additional support of water and food, providing medical personnel and supplies, assisting in the repair of infrastructure, and supplying general clothing and shelter). Each session was guided by one of five facilitators. In addition, one (of four) IS graduate students joined the session as testers to evaluate CoPrA2GO by observing team discussions and performing real-time communication coding. They received a short training on how to use CoPrA2GO and were handed-out an adapted version of the CoPrA code book explaining each code of a communication act. Additionally, testers were asked to write down instant feedback on "what works well", "what does not work well", and "what catches your eye in general". After they finished their coding, they sent the resulting MXML file per email, for backup reasons, and to the CoPrA-Tool test-server, for analyzing it directly.

Additionally, the testers were asked to jointly reflect on their experiences in a focus group interview about their perceptions on the usefulness of CoPrA2GO. The interview lasted for 30 minutes, and it was videotaped and transcribed afterwards. The transcription was analyzed by applying the coding procedure and method of Corbin et al. [22] using ATLAS.ti [23]. This procedure consists of open, axial, and selective coding. The goal of open coding is to break down the data analytically by an interpretive process. This should help to gain new insights on the data and generate subcategories. The goal of axial coding is to relate coding categories to their subcategories and to test these relationships against the data. During this coding phase the categories are also developed further. Finally, the goal of selective coding is to unify all categories, found in the phase before, around a central "core" category, where the core category represents the central phenomenon of the study [22]. Additionally, the sheets for instant feedback were matched with the interview answers to enrich them.

5 Results of the User Acceptance Test

The aim of the qualitative content analysis was to test the user acceptance of the CoPrA2GO application on the basis of three research questions comprising *(1) For which purpose is CoPrA2GO used?*, *(2) What makes CoPrA2GO useable?*, and *(3) In* *which settings could CoPrA2GO be used?* We tested the user acceptance based on the criteria of the technology acceptance model [26], where primarily perceived ease-of-use and perceived usefulness lead to user acceptance of technology.

When addressing research question (1), the results show that three out of four testers see the purpose of CoPrA2GO in providing feedback immediately after the meeting. All of the testers agreed on its usefulness for recognizing team behavior patterns and performing descriptive statistics. In this context, they mentioned that it is very interesting to see how many ideas were generated (descriptive statistics), and which ideas got immediately challenged or supported (team behavior patterns).

"I think that at the end you can easily see who has contributed the most. Especially, the number of generated options of each team member. You can also see the participation of each member." – Tester 1

"Additionally, you are able to see which option is immediately challenged or supported. Therefore, it is nice for pattern recognition." – Tester 2

Testers also mentioned some disadvantages for using CoPrA2GO. The application cannot compensate the reflection cycles that are common to traditional qualitative content coding. That is mainly because the traditional coding of audio or videotaped communication happens at a later point in time, is not as time-pressured, and options for discussing specific codes exists. This is believed to lead to a higher precision of coding.

"It is really fast, but it cannot be compared with manual communication coding, because the level of detail of manual coding cannot be reached. During manual coding you have time to think about the communication log, to discuss about the codes you would like to apply, you are able to perform an intercoder reliability, and more. – [CoPrA2GO] is another approach, a simpler and faster one." – Tester 2

When addressing research question (2), the results show what aspects make CoPrA2GO usable for real-time coding of communication in synchronous, small sized, face-to-face settings. All testers agreed on the clear, understandable, simple, and intuitive interaction with the application and hence described it as an easy to use application.

"It was very easy to interact with the application, the drag-and-drop functionality worked perfectly and the coding never failed." – Tester 4

"[…] The application was simple to use, it was obvious what was to do, and yes, it was very intuitive." – Tester 1

Additionally, the testers agreed on a quick learning phase while using CoPrA2GO. For instance, just one out of four testers had problems with using the application in the first coding session. This could also be seen in the log file where the first session consisted of 20 collaboration act entries with six codes used and the second consisted already of 167 entries with 12 codes used. Thus, he/she needed just 30 minutes to learn how the application should be used.

"In the first session I was not used to the application and the setting, so I was just able to code on a coarse grained level of abstraction. […] In the following sessions I was able to code in much more detail, because I knew the setting of the experiment and I was used to the application." – Tester 1

They also mentioned that it is an interesting coding approach as the efforts inherent to traditional coding are low. With the above outlined constraints, it represents an easy way to code communication.

"An advantage is that you get results without the need of manually coding communication." – Tester 4

When addressing research question (3), testers gave opinions about the setting in which they deem CoPrA2GO as useful. The testers stated that the team's communication needs to be well-structured. The reason for this is that there exist problems with the real-time coding of parallel communication, when just one coder has to code the communication of a small group. Another reason is that the communication should not be too fast-paced, otherwise the coder is too slow to code every aspect of the collaboration session.

"I think that the application is very useful in […] moderated setting, for example, Tester 1, then Tester 3, and finally, I say a sentence." – Tester 2

"To code a discussion after a presentation would be possible." – Tester 4

"Yes, and a podium discussion would be a good example." – Tester 2

"[…] But I could imagine, that in a business meeting, where real discussions happen, it could be very hard to assign the codes [with CoPrA2GO]." – Tester 2

"Yes, because there is parallel communication and discussions are too fast paced to code for one coder. I would propose to use at least two coders for four team members, or for every team member one real-time coder." – Tester 4

Table 1 summarizes the results of the user acceptance test. The first column includes the research question and the second column provides the answers to each research question that were derived from qualitative content coding.

Table 1. Results of the User Acceptance Test

WHAT purposes is CoPrA2GO used for?	• recognizing team behavior patterns • providing quantitative team output statistics
WHAT makes CoPrA2GO usable?	• simplicity • easy-to-learn • no extra manual coding • providing feedback immediately after the meeting
WHERE CoPrA2GO could be used?	• well-structured communication • little parallel communication • limited speed of communication

6 Discussion and Limitations

In this paper we presented CoPrA2GO, an iPad application, for real-time communication coding, which offers the possibility to get feedback on collaboration processes immediately after the session. In the previous section we presented the results of our user acceptance test (summarized in Table 1), which will now be discussed, again structured along the three research questions we posed in the introduction of this paper.

1. For what purposes is CoPrA2GO used?

The user acceptance test showed that the application is a tool that is mainly useful for receiving feedback on team behavior patterns and outputs, and giving the possibility to analyze the generated log files immediately after the meeting. This is possible because each communication act is combined with a timestamp and is stored in chronological order in the communication log, and, in particular, because CoPrA2GO is connected to the CoPrA Tool [12] in the back-end. Additionally, the MXML log files have a structure that can also be analyzed with ProM, which provides additional process metrics that could be applied [20]. Many teams do not have the necessary communication ability to guide their team members or keep their interactions effective [24]. Feedback can mitigate this problem because it allows to draw the team's attention to the task or the group, hence, affecting behavior [25]. Additionally, it is possible to code while being fully aware of the context of the communication and the non-verbal communication in the room, which makes it possible to gain deeper insights in the communication than looking at a simple communication log ex-post. Additionally, CoPrA2GO could also be used to analyze video and/or audiotaped collaboration sessions ex-post. Furthermore, this ex-post approach could be used to refine an initial real-time coding.

2. What makes CoPrA2GO usable?

CoPrA2GO is usable because of its simplicity, clearness, and intuitiveness. The application is easy-to-use and easy-to-learn, because its design is perceived as user-friendly. Specifically, it does not allow erroneous user inputs and offers just the necessary input possibilities. Furthermore, the drag-and-drop functionality is already well established in software, like operating systems and mobile applications, which makes the effort of getting used to it low. The quick learning process is on the one hand influenced by the just mentioned user friendliness and on the other hand by the obviousness of the application usage. Alongside this, coding happens quickly and is relatively effortless, compared to common qualitative coding practices. This is mainly due to the reason that the coding happens on-the-fly during the meeting and is IT-supported, i.e. CoPrA2GO. Additionally, the analysis happens at the back-end with help of the CoPrA Tool which enables CoPrA2GO, unlike other coding systems, to run analysis immediately after coding without extra effort related to the insertion of data in spreadsheets, conversion of log files, or switching consciously between systems.

3. In which settings is CoPrA2GO useful?

The results show that CoPrA2GO is especially a tool for teams having well-structured communication. There were two aspects mentioned, which could hinder the use of the application, namely, parallel and too fast-paced communication. For this reason there is a need of either a facilitator who moderates the collaboration session, or a self-managed team that selects a leader to coordinate their processes. The facilitator's job is to manage the meeting effectively, to handle group dynamics, and to use adequate technology [27]. A facilitator may intervene into the content, process, or how technology should be used [28]. To overcome the issues of parallel and too fast-paced communication, especially, the process facilitation part is interesting. Process facilitation helps a team to manage and coordinate collaboration activities [29]. As a result, the structure of the overall process is improved, for example, by agreeing on interaction routines [28]. Furthermore, this well-structured communication is needed because, according to Cognitive Load Theory [30], the working memory of humans is limited in capacity when it has to process new information. Therefore, it could happen that parallel and too fast-paced communication lead to an overload of cognitive capacity, which can result in a decrease of the overall performance [31]. In our conducted experiment the testers had no issues with cognitive load and, therefore, were capable of keeping up with what was said and of using CoPrA2GO for real-time coding, which was indicated by the user acceptance test. According to the model of technology acceptance [26] it is likely that CoPrA2GO will find acceptance as it is perceived as useful and easy-to-use.

However, there also exist limitations that should be considered. Firstly, the application used during the real-time coding testing scenario is just a stable prototype. Secondly, the number of testers was limited to four, which could lead to a bias in the user acceptance test. Finally, the user acceptance test is based on a focus group interview, where maybe additional single interviews reveal different opinions.

7 Conclusion

This paper introduced CoPrA2GO, an application suitable for real-time coding of communication of small teams in face-to-face settings. The user acceptance test shed light on the perceived purpose of the tool, in which settings it could be used and why its use is perceived as effortless. There exist topics for future research that should be considered. Firstly, the limited code book does not allow tracking socio-emotional cues, such as mood or specific body language. Especially during real-time coding of a collaboration session, the coder has the possibility to see such behavior and assign it to a team member. Also tracking the mood of each team member and the general mood within the team would be an interesting addition for the analysis of team performance. This could be done by adding a mood barometer to the application that reacts interactively when a change in mood is coded. In fact, recent studies [32] demonstrate that it is indeed very difficult to understand individual and team emotions. Even if it is possible to detect emotion of an individual by analyzing video registration it is very difficult, because these emotions are closely related to the actual context that will influence the interpretation of facial and body signals and movements. Therefore, a coder who is present in the collaboration session and aware of the context, is able to detect the general mood and will help to understand also this aspect of collaboration. For this purpose, the MXML schema would also need a revision, adding a specific tag and properties to save mood information. Furthermore, the CoPrA Tool would need new metrics for mood evaluation, and to combine mood with the overall team performance. Secondly, the user acceptance test was performed in small team settings. It would be interesting to validate the usefulness of CoPrA2GO in bigger teams. One challenge could be that a single CoPrA2GO coder might not be able to handle the cognitive load of real-time communication coding of bigger teams.

As result the paper has implications for research and practice. On the one hand, the paper contributes to research because CoPrA2GO represents a tool for IT-supported communication coding with less demand on time and labor than traditional coding system. This should benefit further advances in our research on team effectiveness and collaboration analysis on the basis of communication. On the other hand, it contributes to practice by providing an easy-to-learn and easy-to-use tool for real-time coding and collaboration process feedback immediately after meetings. Consequently, it gives the possibility to get better measurements on teamwork and team performance [11].

Appendix

Table 2. CoPrA2GO Code Book; short forms are in brackets.

Code	***Description***
Information / Knowledge Provision (provision)	Occurs when someone provides clarifications for problem analysis
Information / Knowledge request (request)	Occurs when someone asks for clarifications of the problem analysis, or for repetition of immediately preceding information
Option Generation – Complete (generateOption(c))	Statements explicitly proposing a complete or near complete solution on the basis of parts of solutions and statements that close the discussion on a specific idea
Option Generation – Partly (generateOption)	Statements that provide an incomplete solution and are part of the generation and evolution of an idea
Ask Option (askOption)	Occurs when someone asks for a response to a proposed option which is part of the problem-solving task
Option Challenge (challengeOption)	Occurs when someone provides criticism of a single potential proposed idea (option) in the problem-solving task environment
Option Support (supportOption)	Occurs when someone provides support to a proposed idea, a partly proposed option or to a complete option by providing an argument for the option

Plan Propose (proposePlan)	Utterances that suggest (1) to move on in the team process or (2) to alter the team process by including a further team process step
Plan Support (supportPlan)	Utterances that give reasoning why to support a proposition made for a process or plan regulation
Plan Challenge (challengePlan)	Utterances that give reasoning why to challenge a proposition made for a process or plan regulation
Plan Ask (askPlan)	Question utterances asking where, when, why, who, and how should proceed with the team process
Situation Update (situationUpdate)	Statements that provide information about what the team is currently doing, or what it is currently happening, both on process and task level
Situation Request (situationRequest)	Statements that ask about what the team is currently doing or what is currently happening with the task
Agreement / Disagreement	Expressions of agreement or disagreement with no rationale provided.
Incomplete / Filler (filler)	Utterances that cannot be categorized into one of the other categories because statements are incomplete or just fillers
Non Task (nonTask)	Utterances that signal joking or that are out of the topic of the task
Not Assignable (notAssignable)	In this case no communication action can be associated to a thought unit

References

1. Lawler, E.E., Mohrman, S.A., Ledford, G.E.: Creating High Performance Organizations: Practices and Results of Employee Involvement and Total Quality Management in Fortune 1000 Companies. Jossey-Bass, San Francisco (1995)
2. Kozlowski, S.W.J., Bell, B.S.: Work Groups and Teams in Organizations. In: Borman, W.C., Ilgen, D.R., Klimoski, R.J. (eds.) Handbook of Psychology. Industrial and Organizational Psychology, vol. 12, pp. 333-375. Wiley-Blackwell, New York (2003)
3. De Vreede, G., Briggs, R.: Collaboration Engineering: Foundations and Opportunities. Journal of the Association for Information Systems: Editorial to the Special Issue on the Journal of the Association of Information Systems. 10(3), 121-137 (2009)
4. Pentland, A.: The New Science of Building Great Teams. Harward Business Review. April, 60-70 (2012)
5. Seeber, I., Maier, R., Weber, B.: Macrocognition in Collaboration: Analyzing Processes of Team Knowledge Building with CoPrA. Group Decis Negot. 22(5), 915-942 (2013)

6. Mathieu, J., Maynard, M. T., Rapp, T., Gilson, L.: Team Effectiveness 1997-2007: A Review of Recent Advancements and a Glimpse into the Future. Journal of Management. 34(3), 410-476 (2008)
7. Keyton, J., Beck, S.J: The Influential Role of Relational Messages in Group Interaction. Group Dynamics: Theory, Research, and Practice. 13(1), 14-30 (2009)
8. Bales, R.F.: Interaction Process Analysis: A Method for the Study of Small Groups. Addison-Wesley, Oxford (1950)
9. Poole, M.S., Holmes, M.E.: Decision Development in Computer-Assisted Group Decision-Making. Human Communication Research. 22(1), 90-127 (1995)
10. McCowan, I., Gatica-Perez, D., Bengio, S., Lathoud, G., Barnard, M., Zhang, D.: Automatic Analysis of Multimodal Group Actions in Meetings. IEEE Transactions on Pattern Analysis and Machine Intelligence. 27(3), 305–317 (2005)
11. Salas, E., Cooke, N. J., Rosen, M.a.: On Teams, Teamwork, and Team Performance: Discoveries and Developments. Human Factors: The Journal of the Human Factors and Ergonomics Society. 50(3), 540–547 (2008)
12. Frati, F., Seeber, I.: CoPrA: A tool for Coding and Measuring Communication in Teams. 2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST). 43-48 (2013)
13. Cohen, S.G., Bailey, D.E.: What Makes Teams Work: Group Effectiveness Research from the Shop Floor to the Executive Suite. Journal of Management. 23(3), 239-290 (1997)
14. Bommer, W.H., Johnson, J.L., Rich, G.A., Podsakoff, P.M., MacKenzie, S.B.: On the Interchangeability of Objective and Subjective Measures of Employee Performance. Personnel Psychology. 48(3), 587-605 (1995)
15. Futoran, G.C., Kelly, J.R., McGrath, J.E.: TEMPO: A Time-Based System for Analysis of Group Interaction Process. Basic and Applied Social Psychology. 10(3), 211-232 (1989)
16. Keyton, J., Wall, V.D.: Symlog: Theory and Method for Measuring Group and Organizational Communication. Management Communication Quarterly. 2(4), 544-567 (1989)
17. Myers, M.D.: Qualitative Research in Business & Management. SAGE Publications Limited (2008)
18. Patton, M.Q.: Qualitative Evaluation and Research Methods. Sage Publication. Thousand Oaks, California (2002)
19. Seeber, I., Maier, R., Ceravolo, P., Frati, F.: Tracing the Development of Ideas in Distributed, IT-Supported Teams during Synchronous Collaboration. Proceedings of the 22nd European Conference on Information Systems. Tel Aviv, Israel. 1–17 (2014)
20. van der Aalst, W., van Dongen, B., Günther, C., Mans, R., de Medeiros, A., Rozinat, A., Rubin, V., Song, M., Verbeek, H., Weijters, A.: ProM 4.0: Comprehensive Support for Real Process Analysis. Springer Verlag, Berlin (2007)
21. Hevner, A.R.: A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems. 19(2), 87–92 (2007)
22. Corbin, J., Strauss, A., Clarke, A., Gerhardt, U., Glaser, B.: Grounded Theory Research: Procedures, Canons and Evaluative Criteria. Qualitative Sociology. 13(1), 3-21 (1990)
23. Muhr, T., Friese, S.: User's Manual for ATLAS.ti 5.0. ATLAS. ti Scientific Software Development GmbH. Berlin (2004).
24. Keyton, J., Beck, S. J., Beth, M.: Macrocognition: A Communication Perspective. Theoretical Issues in Ergonomics Science. 11(4), 272–286 (2010)
25. Kluger, A., & DeNisi, A.: The Effects of Feedback Interventions on Performance: A Historical Review, a Meta-Analysis, and a Preliminary Feedback Intervention Theory. Psychological Bulletin. II(2), 254–284 (1996)

26. Davis, F. D.: Preceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly. 13(3), 319–340 (1989)
27. De Vreede, G.-J., Boonstra, J., Niederman, F.: What Is Effective GSS Facilitation? A Qualitative Inquiry into Participants' Perceptions. Proceedings of the 35th Annual Hawaii International Conference on System Sciences, HICSS. 616-627 (2002)
28. Seeber, I., Maier, R., Weber, B.: Opening the Black Box of Team Processes and Emergent States: A Literature Review and Agenda for Research on Team Facilitation. 2014 47th Hawaii International Conference on System Sciences. 473–482 (2014)
29. Tan, B., Wei, K.-K., Lee-Partridge, J.: Effects of Facilitation and Leadership on Meeting Outcomes in a Group Support System Environment. European Journal of Information Systems. 8(4), 233-246 (1999)
30. Sweller, J.: Cognitive Load during Problem Solving. Cognitive Science. 12(2), 257–285 (1988)
31. Paas, F., Renkl, A., Sweller, J.: Cognitive Load Theory: Instructional Implications of the Interaction between Information Structures and Cognitive Architecture. Instructional Science. 32(1/2), 1–8 (2004)
32. Saneiro, M., Santos, O.C., Salmeron-Majadas, S., and Boticario J.G.: Towards Emotion Detection in Educational Scenarios from Facial Expressions and Body Movements through Multimodal Approaches. The Scientific World Journal. Vol. 2014 (2014)