Collaborative Business Process Modeling

Simon Forster

Simon.Forster@uibk.ac.at

Jakob Pinggera

Jakob.Pinggera@uibk.ac.at

Barbara Weber

Barbara.Weber@uibk.ac.at

Abstract: Research on quality issues of business process models has recently begun to explore the process of creating process models. With growing complexity, the creation of business process models requires the presence of several, potentially spatially distributed, stakeholders. As a consequence, the question arises how this affects the process of process modeling. In this paper, we present an extension to Cheetah Experimental Platform, specifically designed for investigating how process models are collaboratively created. Furthermore, we illustrate how various levels of interactions are supported by our modeling editor and outline its usage for analyzing the process of collaboratively creating process models. We believe that this extension is a first step toward understanding the process of process modeling in a collaborative setting which will ultimately help us to develop better process modeling environments.

1 Introduction

"*Business process modeling is the task of creating an explicit, graphical model of a business process from internalized knowledge on that process*" [IRRG09]. The resulting business process models play an important role for the management of business processes [BRU00], depicting how "*various tasks are coordinated to achieve specific organizational goals*" [MDS+10]. Such process models are used to build a consensus among stakeholders involved within the business process [MDS+10]. Therefore, the quality of business process models is essential [KSJ06] as it constitutes a measure of the fulfillment of its purpose (e.g., to serve as a basis for a system development project) [Rit09b]. However, industrial process model collections suffer from a range of quality problems. Understandability of process models suffers from poor quality which subsequently hampers the models' maintainability [Men08, WR08]. Examples for typical quality problems are non-intention-revealing or inconsistent naming [MRR10], redundant process fragments [HBR10] or overly large and unnecessarily complex models [SOM08].

To address these quality problems significant research has been conducted in recent years on factors that impact process model understandability and maintainability [Men08, WR08, KSJ06]. Focus of these works is on the *outcome of the modeling process* [AH00, GL06], i.e., the resulting process model. In turn, relatively little emphasis has been put on the fact that model quality presumably depends upon the modeling process that was followed to create it, i.e., the *process of process modeling* (PPM) [PZW+12]. For example, [PSZ+12] aims at a better understanding of the PPM, i.e., the formalization of a process model from an informal requirements specification. Thereby, [PSZ+12] assumes a modeling setting where a single model engineer is creating a process model and where the communication between model engineers and domain experts is mediated via an informal requirements specification [PZW+12]. However, when looking at the complexity of real life projects it is often not possible to have only a single model engineer creating the corresponding business process model, since knowledge of the business process might be distributed over a number of domain experts [HPvdW05]. Similarly, the corresponding knowledge to create the process model has to be distributed among model engineers. As a consequence, various domain experts and model engineers are involved in the development cycle, who collaboratively create a process model [RKdV08]. By this close collaboration the border between requirements elicitation and formalization becomes blurred. In fact, the distinction between those two phases disappears and is replaced by an iterative process performing them repeatedly.

Even though collaborative process modeling settings are increasingly found in practice [Rit09a, MRW12] and results in software engineering have shown that collaboration can increase quality and efficiency significantly [WKCJ00], the way how process models are collaboratively created is hardly understood [Rit09a]. We want to extend existing work on the PPM, which focuses on single model engineer settings, toward a collaborative setting where multiple stakeholders (e.g., domain experts and model engineers) collaboratively create a process model. Therefore, we developed a modeling tool that enables researchers to record the PPM within collaborative environments as well as the analysis of the data gathered during this process. Our tool not only features a collaborative modeling editor but specifically aims at investigating the PPM to gain an in-depth understanding of the PPM involving multiple stakeholders. Ultimately, we are trying to evaluate whether and to what extent improvements in model quality as well as efficiency can be attributed to collaboration. Therefore, the tool and evaluation techniques presented in this paper serve as starting point for further research in the area of collaborative process modeling. We are planning to conduct case studies and experiments using this tool as the underlying basis.

The remainder of this paper is structured as follows: Sect. 2 presents backgrounds on the PPM and introduces the Cheetah Experimental Platform (CEP) for single model engineer settings, Sect. 3 then details the extensions made to CEP in order to support collaborative process modeling. Related work is discussed in Sect. 4. Finally, Sect. 5 concludes the paper with a summary and outlook on future work.

2 Background

This section provides background information on the PPM and enumerates the individual processes involved (cf. Sect. 2.1). Furthermore, we introduce CEP for single modeler settings (cf. Sect. 2.2).

2.1 The Process of Process Modeling

During the formalization phase process modelers are working on creating syntactically correct process models reflecting a given domain description by interacting with the process modeling tool [HPvdW05]. This modeling process can be described as an iterative and highly flexible process [CWWW00, Mor67], dependent on the individual modeler and the modeling task at hand [Wil95]. At an operational level, the modelers interactions with the tool would typically consist of a cycle of the three successive phases of (1) comprehension (i.e., the modeler forms a mental model of domain behavior), (2) modeling (i.e., the modeler maps the mental model to modeling constructs), and (3) reconciliation (i.e., the modeler reorganizes the process model) [PZW$^+$12, SKW11].

Comprehension. Research on human cognition and problem solving has shed light on comprehension. According to [NS72], when facing a task, the problem solver first formulates a mental representation of the problem, and then uses it for reasoning about the solution and which methods to apply for solving the problem. In process modeling, the task is to create a model which represents the behavior of a domain. The process of forming mental models and applying methods for achieving the task is not done in one step applied to the entire problem. Rather, due to the limited capacity of working memory, the problem is broken down to pieces that are addressed sequentially, chunk by chunk [SKW11, PZW$^+$12].

Modeling. The modeler uses the problem and solution developed in working memory during the previous comprehension phase to materialize the solution in a process model (by creating or changing it) [SKW11, PZW$^+$12]. The modelers utilization of working memory influences the number of modeling steps executed during the modeling phase before forcing the modeler to revisit the problem for acquiring more information [PZW$^+$12].

Reconciliation. After modeling, modelers typically reorganize the process model (e.g., renaming of activities) and utilize the process model's secondary notation (e.g., notation of layout, typographic cues) to enhance the process model's understandability [Pet95, MRC07]. However, the number of reconciliation phases in the PPM is influenced by a modeler's ability of placing elements correctly when creating them, alleviating the need for additional layouting [PZW$^+$12].

2.2 Cheetah Experimental Platform

In order to get a detailed picture of how process models are created, we use Cheetah Experimental Platform (CEP). CEP has been specifically designed for investigating the PPM in a systematic manner [PZW10]. When considering a single modeler interacting with a process modeling environment, the development of process models consists of adding nodes and edges to the process model, naming or renaming these activities, and adding conditions to edges. In addition to these interactions a modeler can influence the process model's secondary notation, e.g., by laying out the process model using move operations for nodes or by utilizing bendpoints to influence the routing of edges (cf. Table 1).

CEP instruments a basic process modeling editor to record each user's interactions together with the corresponding time stamp in an event log, describing the creation of the

User Interaction	Description	User Interaction	Description
CREATE NODE	Create activity or gateway	RENAME	Rename an activity
DELETE NODE	Delete activity or gateway	UPDATE CONDITION	Update an edge's condition
CREATE EDGE	Create an edge connecting two nodes	MOVE NODE	Move a node
DELETE EDGE	Delete edge	MOVE EDGE LABEL	Move the label of an edge
CREATE CONDITION	Create an edge condition	CREATE/DELETE/MOVE EDGE BENDPOINT	Update the routing of an edge
DELETE CONDITION	Delete an edge condition		
RECONNECT EDGE	Reconnect edge from one node to another		

Table 1: User Interactions with CEP

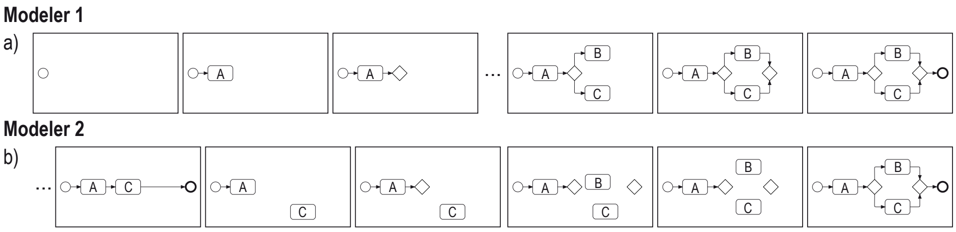

Figure 1: Two Different Processes of Process Modeling to Create the Same Process Model

process model step by step. By capturing all interactions with the modeling tool, we are able to *replay* a recorded modeling process at any point in time without interfering with the modeler or her problem solving efforts. This allows for observing how the process model unfolds on the modeling canvas[1]. Fig. 1 illustrates the basic idea of replaying the creation of a process model. Fig. 1a shows several states of a typical modeling process as it can be observed during replay. Fig. 1b shows the states of a different PPM instance that nonetheless results in the *same* model. This replay functionality of CEP allows to observe in detail *how* modelers create the model on the canvas. Beside the ability of replaying the modeling process, the captured interactions with the modeling tool can be used for evaluation of the PPM. The Modeling Phase Diagram (MPD) is an example of a visualization of the PPM that can be generated using CEP [PZW$^+$12]. This technique classifies the recorded modeling activities according to the cognitive phases introduced in Section 2.1 and provides a graphical presentation for further analysis. A PPMChart is another method illustrating the data captured during the modeling process [CVP$^+$12]. Additionally, CEP provides a calculation extension for various types of metrics [PSZ$^+$12].

3 Collaborative Modeling using CEP

CEP (cf. Sect. 2.2) aims at investigating the process of process modeling within single modeling settings. Here we introduce a tool to analyze the PPM within collaborative modeling settings (cf. Sect. 3.1). Moreover, this section provides a detailed view on extensions necessary to support collaborative business process modeling (cf. Sect. 3.2) as well as extensions required for analyzing the PPM within collaborative modeling settings (cf. Sect. 3.3).

[1] A demonstration of CEP's replay function is available at http://cheetahplatform.org

3.1 Collaborative Processes

When process models are created collaboratively, the individual processes of process modeling (*comprehension*, *modeling* and *reconciliation*) as introduced in Sect. 2 are not sufficient. In addition, team processes take place during which teams exchange information, create solution options, exchange knowledge, evaluate and negotiate alternatives and assess their own processes [FSJS$^+$10]. As a result, the team is building further knowledge and a shared understanding of the process model [FSJS$^+$10, Rit12a]. In order to be able to analyse these processes we extend CEP with support for collaborative modeling. Furthermore, we extend the replay functionality to replay the data retrieved from collaborative features. This, in turn, provides the ability to analyze team processes in detail in combination with the individual processes of single team members.

3.2 CEP Modeler Extension

Extensions to the modeling editor were necessary to enable users to collaboratively and concurrently edit a business process model. In addition, there are a number of levels of social interaction that have to be considered when developing a collaborative modeling tool. According to [MRW12] those levels are *awareness*, *communication*, *coordination*, *group decision making*, and *team-building*.

Awareness. The ability of the participants seeing the same objects as well as the changes done by others. Another example would be a list displaying the names of all participants making participants aware of each other.

Communication. Sending or receiving messages is crucial in case of spatially separated participant. Hence, this level of interaction aims at exchanging messages and establishing a common language [MRW12].

Coordination. As soon as there is more than one person involved in a task, splitting and distributing those tasks is a crucial aspect in collaborative modeling and requires coordination support.

Group decision making. Again, when multiple people are working together, they need a mechanism to propose their solutions to problems. Those solutions can then be evaluated and selected. Meaning the participants are negotiating about the models [Rit07].

Team-building. As a result of collaboratively working together, the team is building further knowledge [FSJS$^+$10] as the participants exchange information between each other.

Fulfilling those levels of interaction is essential for effectively supporting collaborative process modeling.

3.2.1 Collaborative Modeling Support

The foremost level of interaction is the *awareness* level as it is crucial for spatially separated domain experts and model engineers to be able to graphically create process models in a collaborative manner [MRW12]. Participants are able to see changes made to the process model immediately. As an example, Fig. 2 illustrates participants using the collaborative modeling editor. Participants *Alice* and *Bob* are working on the same model.

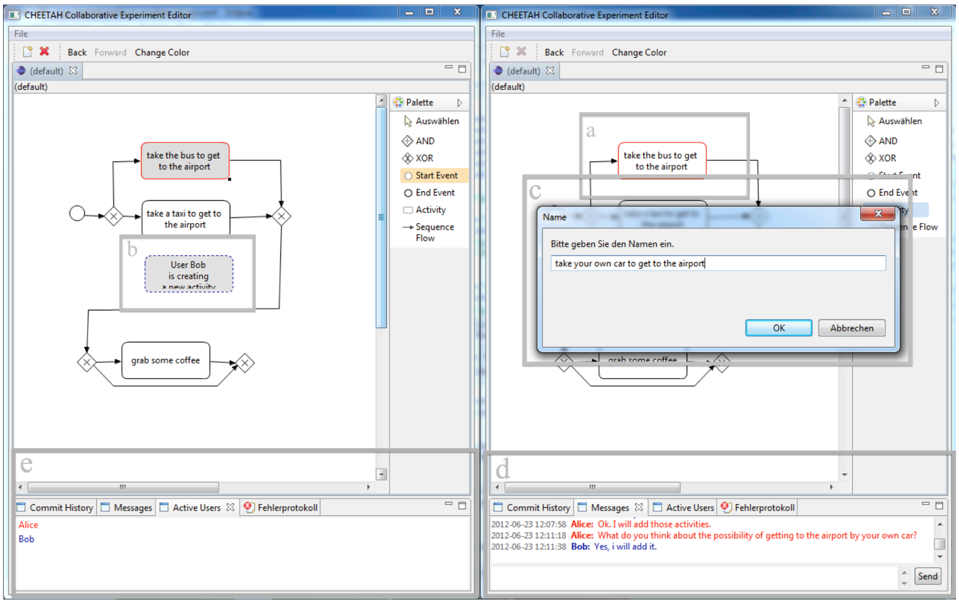

Figure 2: Participants using the Collaborative Process Modeling Editor for Working on the Same Model

Hence, they can see the same process model on their respective canvases. To further increase change awareness participants are supported to identify on which parts of the model other participants are currently working. For this purpose, activities currently selected or moved by other participants are highlighted (cf. Fig. 2a) using colors. Therefore, a unique color is linked to each participant. Meaning, at any time during the modeling process, participants are not only able to see which activities are edited by other participants, but they can also see which participants are working on specific activities. In addition, the tool is visualizing *if* and *who* is currently creating a new activity using colors. Creating an activity is a task of several seconds as the participant (*Bob*) has to enter a name for the activity (Fig. 2c). In order to increase change awareness when creating a new activity the tool already displays the new activity (cf. Fig. 2b) on the canvases of the other participants (*Alice*). Until the participant has entered the name of the activity, the other participants can see the activity displaying the name of the participant creating that activity. This way, the coordination of creating new activities can be eased.

3.2.2 Communication Support

As already mentioned, within a collaborative modeling environment participants are spatially separated from each other. Since *communication* is another important level of interaction we integrated a communication mechanism into CEP. More precisely, we integrated a “chat window” into CEP. With the help of this feature the different participants taking part in the modeling process can exchange knowledge regarding different aspects of the model, create solution options, evaluate and negotiate alternatives [Rit07], and assess their

User Interaction	Description
Join	Join a modeling session
Leave	Leave a modeling session
Message	Send a message
Change	Send a modeling interaction (e.g., CREATE NODE, CREATE EDGE, RENAME)
Conflict	Indicate a conflicting model interaction (e.g., CREATE NODE, CREATE EDGE, RENAME)

Table 2: User Interactions with Collaborative CEP

own process [FSJS$^+$10]. Hence, this feature additionally aims at supporting *coordination* and *group decision making*.

When developing this feature we also considered awareness support. For this, messages exchanged between the participants are highlighted in the colors linked to participants within the chat window (cf. Fig. 2d). In addition, to facilitate communication and to foster awareness between modeling participants, we added another window which displays all currently connected participants (cf. Fig. 2e).

Using those features our tool aims at the fulfillment of each level of interaction (i.e., *awareness*, *communication*, *coordination* and *group decision making*) according to [MRW12]. As a result, our tool supports the *team-building* process within the collaborative PPM as it provides a formal (i.e., modeling editor) as well as an informal (i.e., communication window) way interacting with other participants [MRW12].

3.2.3 Logging Extensions

Each interaction with the modeling editor is automatically tracked and logged by our tool resulting in revision control of the commands sent. Using this revision control the modeling editor also provides conflict management preventing race conditions. In case, two conflicting commands (e.g., deleting and moving the same activity) are sent to the server at the same time, the first client trying to execute the conflicting command (moving the activity) recognizes the problem and marks it being a conflict. As a result, the other clients will not execute the conflicting command. Like the actual modeling process (e.g., creation/deletion of activities or edges) also the messages sent (cf. Sect. 3.2.1) are tracked by our prototype. Moreover, the messages are automatically linked to the activities selected during creation of the message. This information, including timestamp and information regarding the user, is important for later analysis of the modeling process (cf. Sect. 3.3.2). Additionally, the modeling events logged within CEP are extended. In order to be able to retrieve the user's data performing modeling commands, we extend the data model of CEP. Therefore, we wrap the additional information captured with our tool around the modeling commands already logged by CEP (cf. Tab. 1). Tab. 2 lists the commands created and exchanged by collaborative CEP.

The modeling commands of CEP (cf. Tab. 1) are wrapped into the *Change* command of collaborative CEP (cf. Tab. 2) and stored within a data model (cf. Tab. 3).

Hence, after the modeling session ended, it can be identified *who* changed *which* elements

Attribute	Description
Revision-ID	Unique identifier of the command
User	User executing the command
Time	Timestamp of the command
Type	Type of the command (i.e., *Join*, *Leave*, *Message*, *Change*, *Conflict*).
Command	The user interaction

Table 3: Data Model of Collaborative CEP

as well as *when* these elements were changed. In addition, it can be analyzed which messages were exchanged.

3.3 Analysis Extension

Beside the extensions of the model editor (cf. Sect. 3.2) also the analysis capabilities of CEP have to be extended (cf. Sect. 3.3.1) in order to track and evaluate the team processes during a collaborative modeling session and to enable an integrated analysis with the individual processes (cf. Sect. 3.3.2).

3.3.1 Replay Functionality

Using CEP as a grounding makes it possible to use the built in "replay functionality" (cf. Sect. 2.2) which allows replaying process models created with the CEP modeler. In order to support the integrated replay of how the model was created, including the communication which took place, we extend the replay functionality of CEP. Our model editor not only records the modeling steps, but also communication between participants which can later be chronologically reviewed and evaluated.

Again, having usability and the corresponding *awareness* level in mind, we made this feature available for participants. This way, participants joining the modeling session at a later point in time have the opportunity to chronologically recap the evolution of the business process model in terms of activities created as well as messages exchanged yielding a deeper understanding of the process and increasing the awareness of which changes have occurred since they left the modeling session. This feature cannot only be used by participants joining a session later, but at all stages during the modeling process in case participants want to recap why a specific element was modeled this way.

This functionality is illustrated in Fig. 2. The right instance depicts the latest version of the model whereas the left instance shows the model at an earlier point in time. Unless participants are not on the latest version, the modeling window is locked, meaning the participants are not able to interact with the model (e.g., create new activities) or other participants using the chat window. Interacting with the model again is only possible as soon as the participants are at the latest version of the model.

3.3.2 Metrics

After capturing the PPM within a collaborative environment the next step is the evaluation of the retrieved data. More precisely, team processes can be analyzed in addition to the

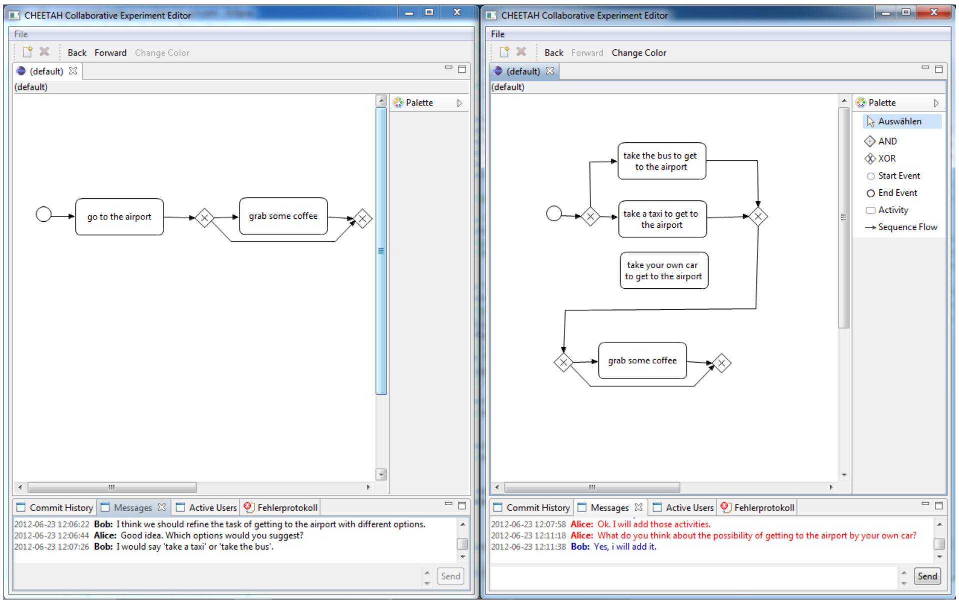

Figure 3: Two Instances Working on the Same Model at Different Points in Time

individual processes [PZW+12]. In order to make qualified assertions, we will develop visualizations, algorithms and metrics to analyze the collaborative modeling process. In particular, we present metrics for different perspectives. Here, the *model perspective* examines the modeling aspects themselves. For example, this perspective comprises measures like number of changes per activity potentially indicating modeling elements which caused difficulties during the modeling process or which caused lots of controversy. Whereas, investigating the participation and collaboration of the team members is in the focus of the *team perspective*. Combining those two perspectives results in the *integrated perspective*. This perspective investigates the interaction of the participants with the model.

“Number of changes per activity” (cf. Tab. 4) can be a measure for the difficulty of a single model element whereas the two team perspective metrics can be a measure for the participants providing the most domain knowledge. In order to be able to filter out valuable comments, we will be utilizing the CoPrA method for analyzing conversation protocols [SWM12]. Moreover, the metrics for the integrated perspective are a measure for the involvement of the participants as well as the importance of single activities. Here, “number of changes per activity by different participants” and “number of comments per modeling element” could indicate activities where the participants had to find a consensus during the modeling phase. Knowledge transfer as well as improved quality are positive results of this negotiation phase. “Number of activities created per participant” and “number of changes per participant” could expose participants claiming the leading role during the modeling process.

For the purpose of visualization, heat maps will be utilized. For example, related to metric

Perspective	Possible metrics
Model perspective	Number of changes per activity
Team perspective	Number of comments made per participant Number of times a critique is offered for a particular modeling element
Integrated perspective	Number of activities created per participant Number of changes per participant Number of changes per activity by different participants Number of comments per modeling element

Table 4: Metrics for the evaluation of the modeling process

"number of comments per modeling element" (cf. Tab. 4) modeling elements could be highlighted in different colors depending on the number of comments. Elements discussed a lot would then appear darker than less discussed ones.

The data needed for those metrics can be generated automatically from the information logged by CEP (cf. Sect. 3.2.3) because the name of the participant is assigned to each modeling command performed within CEP. Also, the name of the participant is linked to the message. A semiautomatic algorithm for linking comments with the corresponding model elements is under development, supporting the subsequent coding of conversation protocols (e.g., using the negotiation patterns [Rit07]). Still, there are metrics (e.g., number of times a critique is offered for a particular modeling element) where it is not possible to generate the needed data automatically out of the information logged. As already mentioned, evaluating those data will be done using the CoPrA method [SWM12]. The resulting data can then be utilized for statistical analysis.

3.3.3 Experimental Workflow

Ultimately, the goal will be performing case studies as well as conducting controlled experiments using this tool. Such experiments consist of a series of tasks that have to be executed by the experiment's subjects, referred to as *Experimental Workflow* [PZW10]. As the collaborative modeling editor builds upon CEP which includes the ability of incorporating components commonly used in experiments (e.g., questionnaires, surveys, tutorials) into controlled experiments, we can make use of this infrastructure. This way we can combine the automatically collected data regarding the process of process modeling with information provided by the modeler before or after the modeling session (e.g., demographic information, domain knowledge, process modeling knowledge, perceived quality, perceived ease of use, perceived usefulness, mental effort).

4 Related Work

This section presents related work in the area of collaborative process modeling.

Research on the Process of Process Modeling. The PPM is concerned with the interaction of the participants (e.g., domain experts and model engineers) during modeling. How important the modeling process itself really is beside the actual outcome is stated in [HPvdW05]. There has already been some research on the PPM [PZW+12, PSZ+12]. However these works focus on modeling settings where a single model engineer creates the process model, whereas with our tool it is possible to investigate how collaborative process modeling impacts the process of creating process models. As an exception, [Rit07] also investigates on collaborative modeling settings concentrating on the negotiation phase of this process. In addition, the team processes (e.g., combination of best performing teams) are investigated in [Rit12a] and evaluated in respect to model quality. Again, with our tool it is possible to analyze the *process of process modeling* and investigate the team processes in addition to the individual processes.

Alternative Process Modeling Tools. There already exist some environments fostering collaboration between different stakeholders. One example of such an environment is Collaborative Modeling Architecture [Rit10]. The COMA Tool provides process model collaboration by means of negotiation on proposed models by the participants.

In contrast to this collaboration methodology, where the participants work in an asynchronous way there also exists another one, synchronous collaboration or concurrent modeling where the participants are working synchronously together on the same model. The advantage of this approach is the fact that participants are able to track model changes immediately. Examples are the Signavio Process Editor[2] and the Software AG's ARIS[3] collaborative tool where it is possible to work simultaneously together on one model using a web browser. CoMoMod [DHFL11] is another example of a collaborative modeling tool. Beside the lack of supporting the BPMN process modeling notation it aims at the modeling outcome rather than the modeling process itself. The same holds for the Signavio Process Editor as well as the Software AG's ARIS collaborative tool.

Here, we do not want to create an alternative but the opportunity to analyze collaboration in a controlled manner. With the possibility of tracking the modeling process, CEP is an ideal platform for this purpose.

Research on Collaborative Process Modeling. There has already been some research in the area of collaborative process modeling [Rit12b, Rit12a]. The team-building processes when creating a model collaboratively using a proposal based tool (COMA) and allowing face to face communication are investigated in [Rit12b] and evaluated in respect to model quality. Again, using our tool it is possible to analyze the PPM. Furthermore, our tool provides the possibility of synchronously working on the same model while being spatial separated using an integrated communication channel.

Research on Process Model Quality. Different research has already been done in the area of business process model quality [Rit09b]. Additionally, there exist guidelines describing quality considerations for business process models [BRU00], the Seven Process Modeling Guidelines (7PMG) defining desirable characteristics of a business process model [MRvdA10] or identifying various aspects of process models' quality [KSJ06]. [MRC07] investigated the influence of model complexity on process model understandability. Prediction models for usability and maintainability for process models are provided by [RSG+09]. The impact of different quality metrics on error probability was

[2]http://www.signavio.com/
[3]http://www.softwareag.com/

investigated in [MVvD$^+$08, Men08]. The role visual notations are playing for cognitive effectiveness is discussed in [Moo09]. The commonality of those works is the focus on the resulting process model, whereas only little attention is payed on the process of modeling itself.

5 Summary and Outlook

This paper introduced a tool to support both, collaboratively creating a business process model as well as analyzing the *process of process modeling* within a collaborative modeling setting. Therefore, we extended Cheetah Experimental Platform with collaboration features like concurrent modeling and a communication channel. Furthermore, we introduced metrics for the evaluation of the data captured by the tool. Therefore, the collaborative modeling editor logs each user interaction for later analysis. The integrated replay functionality not only fosters the evaluation of the PPM but also yields a deeper understanding of the process and increases the awareness of which changes occurred since participants left the modeling session.

After extending CEP to support collaborative process modeling settings we will next perform an exploratory study. The information retrieved in this study will be used for further improving the collaborative modeling editor. Afterwards, we will conduct controlled experiments using this tool. The data obtained in those experiments will then be evaluated using the visualizations and metrics introduced in Sect. 3.3.2.

Further, we plan to extend our prototype with additional features to increase the usability of the tool. An example would be the integration of speech. This would complement the chat window with a convenient way of communicating with other participants and approaches a face-to-face interaction.

Acknowledgements. This research was funded by the Austrian Science Fund (FWF): P23699-N23. Particularly, we would like to thank Peter Zangerl for his developing efforts.

References

[AH00] W.M.P. van der Aalst and A.H.M. ter Hofstede. Verification of Workflow Task Structures: A Petri-net-based Approach. *Information Systems*, 25(1):43–69, 2000.

[BRU00] Jörg Becker, Michael Rosemann, and Christoph Uthmann. Guidelines of Business Process Modeling. In *Business Process Management, Models, Techniques, and Empirical Studies*, pages 30–49, London, UK, 2000. Springer-Verlag.

[CVP$^+$12] Jan Claes, Irene Vanderfeesten, J. Pinggera, H.A. Reijers, B. Weber, and G. Poels. Visualizing the Process of Process Modeling with PPMCharts. In *Proc. TAProViz '12 (accepted)*, 2012.

[CWWW00] Andrew W. Crapo, Laurie B. Waisel, William A. Wallace, and Thomas R. Willemain. Visualization and the process of modeling: a cognitive-theoretic view. In *Proc. KDD '00*, pages 218–226, 2000.

[DHFL11] Thorsten Dollmann, Constantin Houy, Peter Fettke, and Peter Loos. Collaborative Business Process Modeling with CoMoMod - A Toolkit for Model Integration in Distributed Cooperation Environments. In *Proc. WETICE '11*, pages 217–222, 2011.

[FSJS+10] Stephen M. Fiore, Kimberly A. Smith-Jentsch, Eduardo Salas, Norman Warner, and Michael Letsky. Towards an understanding of macrocognition in teams: developing and defining complex collaborative processes and products. *Theoretical Issues in Ergonomics Science*, 11(4):250–271, 2010.

[GL06] Volker Gruhn and Ralf Laue. Complexity metrics for business process models. In *Proc. BIS '06*, pages 1–12, 2006.

[HBR10] Alena Hallerbach, Thomas Bauer, and Manfred Reichert. Capturing Variability in Business Process Models: The Provop Approach. *Journal of Software Maintenance*, 22(6-7):519–546, 2010.

[HPvdW05] S.J.B.A. Hoppenbrouwers, H.A. Proper, and Th.P. van der Weide. A Fundamental View on the Process of Conceptual Modeling. In *Proc. ER '05*, pages 128–143, 2005.

[IRRG09] Marta Indulska, Jan Recker, Michael Rosemann, and Peter F. Green. Business Process Modeling: Current Issues and Future Challenges. In *Proc. CAiSE*, pages 501–514, 2009.

[KSJ06] John Krogstie, Guttorm Sindre, and Havard Jorgensen. Process models representing knowledge for action: a revised quality framework. *EJIS*, 15:91–102, 2006.

[MDS+10] Debdoot Mukherjee, Pankaj Dhoolia, Saurabh Sinha, Aubrey J. Rembert, and Mangala Gowri Nanda. From Informal Process Diagrams to Formal Process Models. In *Proc. BPM '10*, pages 145–161. Springer, 2010.

[Men08] Jan Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction and Guidelines for Correctness*. Springer, 2008.

[Moo09] Daniel L. Moody. The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35:756–779, 2009.

[Mor67] William T. Morris. On the Art of Modeling. *Management Science*, 13(12):B–707–B–717, 1967.

[MRC07] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. What Makes Process Models Understandable? In *Proc. BPM '07*, pages 48–63, 2007.

[MRR10] Jan Mendling, Hajo A. Reijers, and Jan Recker. Activity Labeling in Process Modeling: Empirical Insights and Recommendations. *Information Systems*, 35(4):467–482, 2010.

[MRvdA10] Jan Mendling, Hajo A. Reijers, and Wil M. van der Aalst. Seven Process Modeling Guidelines (7PMG). *IST*, 52(2):127–136, 2010.

[MRW12] Jan Mendling, Jan C. Recker, and Johannes Wolf. Collaboration features in current BPM tools. *EMISA Forum*, 32(1):48–65, 2012.

[MVvD+08] Jan Mendling, H.M.W. Verbeek, Boudewijn van Dongen, W.M.P. van der Aalst, and Gustav Neumann. Detection and Prediction of Errors in EPCs of the SAP Reference Model. *DKE*, 64(1):312–329, 2008.

[NS72] A. Newell and H.A. Simon. *Human problem Solving*. Prentice Hall, 1972.

[Pet95] Marian Petre. Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming. *Commun. ACM*, pages 33–44, 1995.

[PSZ+12] J. Pinggera, P. Soffer, S. Zugal, B. Weber, M. Weidlich, D. Fahland, H.A. Reijers, and J. Mendling. Modeling Styles in Business Process Modeling. In *Proc. BPMDS '12*, pages 151–166, 2012.

[PZW10] Jakob Pinggera, Stefan Zugal, and Barbara Weber. Investigating the Process of Process Modeling with Cheetah Experimental Platform. In *Proc. ER-POIS'10*, pages 13–18, 2010.

[PZW+12] Jakob Pinggera, Stefan Zugal, Matthias Weidlich, Dirk Fahland, Barbara Weber, Jan Mendling, and Hajo A. Reijers. Tracing the Process of Process Modeling with Modeling Phase Diagrams. In *Proc. ER-BPM '11*, pages 370–382, 2012.

[Rit07] Peter Rittgen. Negotiating Models. In *Proc. CAiSE '07*, pages 561–573, 2007.

[Rit09a] Peter Rittgen. Collaborative Modeling – A Design Science Approach. In *Proc. HICSS '09*, pages 1–10, 2009.

[Rit09b] Peter Rittgen. Collaborative modeling of business processes: a comparative case study. In *Proc. SAC '09*, pages 225–230, 2009.

[Rit10] Peter Rittgen. Collaborative Business Process Modeling Tool Support for Solving Typical Problems. In *Proc. Conf-IRM '10*, 2010.

[Rit12a] Peter Rittgen. End-User Involvement and Team Factors in Business Process Modeling. In *Proc. HICSS '12*, pages 180–189, 2012.

[Rit12b] Peter Rittgen. The role of editor in collaborative modeling. In *Proc. SAC '12*, pages 1674–1679, 2012.

[RKdV08] Michiel Renger, Gwendolyn L. Kolfschoten, and Gert-Jan de Vreede. Using Interactive Whiteboard Technology to Support Collaborative Modeling. In *Proc. CRIWG '08*, pages 356–363, 2008.

[RSG+09] Elvira Rolon, Laura Sanchez, Felix Garcia, Francisco Ruiz, Mario Piattini, Danilo Caivano, and Giuseppe Visaggio. Prediction Models for BPMN Usability and Maintainability. In *Proc. CEC '09*, pages 383–390, 2009.

[SKW11] Pnina Soffer, Maya Kaner, and Yair Wand. Towards Understanding the Process of Process Modeling: Theoretical and Empirical Considerations. In *Business Process Management Workshops*, pages 357–369. Springer, 2011.

[SOM08] Martín Soto, Alexis Ocampo, and Jürgen Münch. The Secret Life of a Process Description: A Look into the Evolution of a Large Process Model. In *Proc. ICSP '08*, pages 257–268, 2008.

[SWM12] Isabella Seeber, Barbara Weber, and Ronald Maier. CoPrA: A Process Analysis Technique to Investigate Collaboration in Groups. In *Proc. HICCS '12*, pages 363–372, 2012.

[Wil95] Thomas R. Willemain. Model Formulation: What Experts Think about and When. *Operations Research*, 43(6):916–932, 1995.

[WKCJ00] Laurie Williams, Robert R. Kessler, Ward Cunningham, and Ron Jeffries. Strengthening the Case for Pair Programming. *IEEE Software*, 17(4):19–25, 2000.

[WR08] Barbara Weber and Manfred Reichert. Refactoring Process Models in Large Process Repositories. In *Proc. CAiSE '08*, pages 124–139, 2008.