When to Say "Enough is Enough!": A Study on the Evolution of Collaboratively Created Process Models

IRENE-ANGELICA CHOUNTA, Carnegie Mellon University
ALEXANDER NOLTE, University of Pittsburgh & Carnegie Mellon University
TOBIAS HECKING, University of Duisburg-Essen
ROSTA FARZAN, University of Pittsburgh
THOMAS HERRMANN, Ruhr-University Bochum

ABSTRACT

Organizations conduct series of face-to-face meetings aiming to improve work practices. In these meetings, participants from different backgrounds collaboratively design artifacts, such as knowledge or process maps. Such meetings are orchestrated and carried out by facilitators and the success of the meetings almost solely depends on the experience of the facilitators. Previous research has mainly focused on approaches that support facilitators and participants in the upfront planning of such events. There is however, little guidance for facilitators and participants once a meeting has started. One critical aspect – among others – is that during a meeting, the facilitator and participants need to decide for how long the iterative process of discussion and design should continue. We argue that we can provide support for such decisions based on the evolution of artifacts collaboratively created during such meetings. This paper presents a multi-level, multi-method analysis of artifacts based on experts' observations in combination with network analytics. We study the use of automated analytics to assess the evolution of collaboratively created artifacts and to indicate maturity and established consensus of the collaborative practice. We propose a computational approach to support facilitators and participants in deciding when to stop face-to-face meetings.

CCS Concepts: • **Human-centered computing~Collaborative interaction** • **Human-centered computing~Computer supported cooperative work** • *Human-centered computing~Empirical studies in collaborative and social computing*

Author Keywords

CSCW; collaborative modeling; analytics; modeling workshops; network analysis.

ACM Reference format:

2017. When to Say "Enough is Enough!": A Study On the Evolution of Collaboratively Created Process Models. *Proc. ACM Hum.-Comput. Interact.* 1, 2, Article 33 (November 2017), 20 pages.
DOI: 10.1145/3134668

33

Authors' email addresses are: ichounta@cs.cmu.edu, anolte@pitt.edu, hecking@collide.info, rfarzan@pitt.edu, thomas.herrmann@rub.de

2573-0142/2017/November – Article 33…$15.00

https://doi.org/10.1145/3134668

INTRODUCTION

It is common practice for organizations to arrange meetings where stakeholders collaboratively analyze processes for various purposes: documentation and improvement of current work practice, requirements specification for software development, workflow management, knowledge management and others [36]. Companies invest significantly in such activities. As indicated by a bi-annual survey conducted by Harmon and Wolf since 2005, companies spent from USD 500,000 to over USD 50 million annually on process management [18]. Business process management involves the creation of graphical models (Figure 1) to represent business or work processes [21]. These process models are created collaboratively in face-to-face, multi-party workshops to ensure that multiple perspectives will be taken into account. Process models are created based on a modeling notation and they consist of graphical shapes combined with textual descriptions [15]. There is a variety of process modeling notations such as UML [33], BPMN [34], SeeMe [22] and others. Modeling notations partially overlap but include different visual elements that allow for addressing different aspects of process management. However all process modeling notations make use of the basic building blocks of processes: activities, actors, resources, decisions and flow [15].

Figure 1 shows an example of a process model created in SeeMe. The model consists of the basic building blocks of processes: activities (yellow), actors (orange), resources (blue), decisions (green) and arrows for flow and for relations between activities, actors and resources. This example depicts a process in which custom software is built for a specific customer. The process starts with a phase in which the customer and a consultant jointly develop a requirements document. In the next step, software engineers in collaboration with the consultant use this document to create a prototype of the software. Subsequently, the software engineers implement the software based on the prototype and they test it along with the customer. The process model depicts this workflow as a sequence of individual phases that involve different actors who need to collaborate. The backward arrows of the process model indicate that it is possible to go back to one of the previous phases if the involved actors decide this is required (green decision symbol). Process models like the one in Figure 1 are used to analyze processes, identify problems and develop means to subsequently improve the process [15]. They are also used to build a common understanding about the way people (e.g. in this case customers, consultants and software engineers) collaborate and thus serve as a means to build common ground [12] among process stakeholders and to reach a decision (for example, to decide how to change a process in the future).

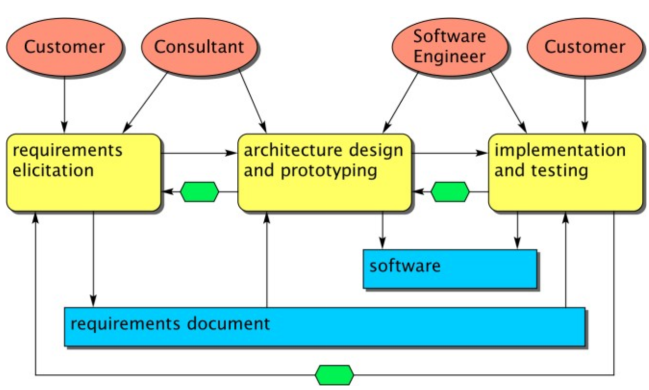

Figure 1. Example process using the SeeMe modeling notation [20]. The different colored shapes represent actors (orange), activities (yellow), resources (blue) and decisions (green). The arrows described the relationships between elements.

Collaboration in these settings is supported by facilitators. Facilitators guide the communication and translate verbal contributions of participants into elements of a process modeling notation [35]. There are various approaches that propose ways for facilitators to conduct workshops [15,21,37]. For example, some approaches provide guidelines for the facilitator on how to lead a discussion (for example, on the design of a newly created software) that will end in an actionable concept [3,8,21]. During such discussions, the facilitator has to make sure that all participants are heard, their views are represented, and the participants share a common understanding. Furthermore, the facilitator helps to establish consensus about the subject matter of the discussion.

While such guidelines are useful for preparing workshops, there is little or no support for both, facilitators and participants, during the course of a workshop. We suggest that we can use automated analytics that capture the evolution of collaboratively created artifacts – such as the process models described above – to provide feedback to facilitators and participants and to inform important decisions. To provide such computational support, it is important to identify automated metrics to operationalize the evolution of the modeling process over time. To that end, we propose the use of network analytics since the evolution of a process model can be perceived as a dynamic network that portrays collaborative practice. As an initial proof of concept, we will apply such analytics to a common decision that most facilitators and participants face: whether to invest time and effort to continue with another modeling workshop or to declare a modeling process complete. Completeness of a model means that it is sufficiently mature to serve as a plan for further action, for instance for process improvement. We acknowledge that there are many other interesting questions that could be asked in this particular context. However, we argue that the question of when to discontinue modeling is important for three reasons:

a) Process models are usually created with a specific aim such as coming up with requirements for a new software tool or with ways of (re-) designing work and business processes. There is thus a necessity to identify a point when a theoretical analysis based on a process model has reached the limits of gaining useful insights, and it is required to try out the results of that analysis in practice;

b) The question of real-time guidance with respect to decision-making on how to proceed during a series of meetings is not sufficiently addressed in literature. Once the workshop has started, the facilitators and participants are left to themselves. This can become an issue especially for less experienced facilitators since the result of a workshop strongly depends on their skills [7];

c) We perceive this question as an initial application for the use of analytics in the context of collaborative workshops in which artifacts are jointly created.

Our results suggest that analyzing the evolution of the process models can serve as a means to support the decision of turning from discussion towards application. The research objective of this paper is twofold. First, it explores the connection between the evolution of collaboratively-created artifacts and the process of collaborative design. Second, we propose a computational approach to provide feedback and support to facilitators and participants during modeling workshops. We argue that this approach can potentially be generalized and used for other process modeling notations and other workshops in which graphical artifacts are created collaboratively, such as concept and decision maps. It should be noted that our approach does not aim in any way at taking control away from facilitators and participants. The goal is rather to provide feedback to both groups to support their decision making.

RELATED WORK

Workshops in which participants collaboratively work on artifacts to analyze and (re-)design work and business processes are a common activity [35,37]. The resulting product – the graphical representation of a process in a process model using a process modeling notation [15] – is one clear focus of such workshops. However, artifacts created in such settings do not only serve as a documentation of workshop outcomes. They are also used as a means to support building a common ground and understanding among workshop participants [12]. This is crucial since collaborative process modeling requires discussing and understanding a process from different perspectives based on different experiences and backgrounds with individuals from various domains. Therefore, artifacts such as process models serve as a basis to share knowledge and experiences [1]. They also support participants to cope with the complexity of real world phenomena [4] by allowing them to decompose the complexity of real work processes and visualize those components and their relationships [31]. Finally, process models also serve as a means to discuss issues related to a process and thus can be perceived to support the development of consensus among workshop participants [38].

Workshops in this context are commonly divided into different phases [3,23] starting with a phase during which information is collected about process steps [24] as well as actors carrying out those steps, resources that are required and decisions that have to be taken. In the later phases, the collected parts of a process are aligned to each other using relations (arrows in Figure 1). Thus, the first phases focus more on gathering elements while the latter phases focus more on process model refinements such as adding details. Phases can last over multiple workshops and can be repeated if the facilitator or the participants deem necessary. The goal at the end is to arrive at a high-quality process model that all participants agree on and that is considered as complete since it represents their common understanding about the process that is visualized.

Despite the previously described benefits, designing artifacts in workshops collaboratively also poses a number of limitations. Workshops are hard to organize due to time constraints and availability of participants. They are also oftentimes perceived as time consuming, stressful and frustrating [39]. In addition, research has established that their outcome strongly depends on the experience of the facilitator [7,19] which has led to attempts to develop patterns that support less experiences facilitators in conducting workshops [26]. Those approaches however focus on providing scripts based on situational descriptions and based on the goals of a workshop. They do not provide feedback to facilitators and participants about the current state of collaboration. In addition, it has to be noted that workshops are usually only a part of processes (re-)design since modifications to processes have to be tested in real life situations in order to assess the effectiveness of the changes and potentially make corrections [15]. Therefore, it is common to conduct a series of workshops before evaluating a design and come back to conducting additional workshops in which evaluation results are discussed and the design is potentially adapted. One of the central questions that participants and facilitators face in this context is **whether or not it is necessary or reasonable to conduct more workshops**. We argue that the evolution of collaboratively created artifacts – such as process models – can serve as an indicator for collaboration progress and thus they can be used to inform the decision to continue conducting workshops.

There are a number of different approaches to assess the quality of a process model. These cover three main perspectives [27]: syntactic, semantic and pragmatic quality. The syntactic quality describes whether or not a process model is correct with respect to the grammar of the process modeling notation. The semantic quality describes how well a process model depicts the real-world process it is supposed to depict. The pragmatic quality describes whether a process model can be easily understood in relation to its complexity. The syntactic quality of a process model can be automatically assessed through process model checking while the semantic and pragmatic quality require the assessment by process modeling experts [27]. The latter qualities can also be assessed through participant feedback [42] as well as through complexity metrics [5] since it has been shown that the complexity of a process model strongly correlates with its understandability [32]. None of these approaches however takes the evolution of process models into account. There are – to the best of our knowledge – no approaches that assess the evolution of process models with the goal to provide feedback to facilitators and participants.

Understanding the evolution of a process model can be essential to provide feedback to all involved participants on their progress towards an improved process model and how close they are to a rather complete process model. The questions of the extent of progress and of how close a process model is to completion are hard to answer. Being able to answer these questions, however, is of strong relevance for both stakeholders and process modeling experts to support their decision about continuing or terminating conducting future workshops. Being able to make an informed decision regarding the progress of the process model can save time, money, and effort for stakeholders and facilitators. To tackle this challenge, we propose the use of metrics of activity and interaction which are quite popular in technology-enhanced learning and educational data mining. In particular, such kind of metrics are used for mirroring, monitoring and guiding purposes, i.e. to assess the evolution of the learning process and to provide feedback to learners and to instructors for reflection and improvement [41]. These metrics usually derive from data-driven approaches and represent activity volume (such as sum of messages, average number of words per message) [16] or more advance measures such as symmetry of contribution [30], temporal proximity [10], etc.

In this work, we explore the use of dynamic networks to model the evolution of process models over time. The evolution of many real-world networks does not follow a random model and there are mechanisms that rule the way a network structure develops. Therefore, the properties of a network can potentially provide meaningful insights. Dynamic graphs and their time-evolution properties are studied across various domains. The Erdös Rényi model [17] describes the evolution of graphs when the likelihood of a connection between any two nodes is uniformly distributed and the degree distribution follows a Poisson distribution. In this case, the average path length of the network tends to decrease when new edges are added to the network [44]. However the degree distribution often follows an inverse power law instead of a Poisson distribution for real-world networks [6]. One model that explains the emergence of scale-free networks is the preferential attachment model [6]. According to this model, nodes will more likely connect with nodes that already have many connections. Leskovec et al. [29] investigated patterns of the properties of dynamic graphs over time and showed that real-world graphs become more dense as they grow, and also their effective diameters decrease in some cases. We explore the use of network properties and structural characteristics, such as diameter, average path length and density. Such metrics have been used to evaluate the communication and knowledge flow between users that engage in learning activities involving the collaborative construction of diagrammatic representations [11,25].

METHOD OF THE STUDY

Study and Dataset Description

In this study, we analyzed 35 process models that were created during a 3-year span within the same organization (a large academic institution). We analyzed a wide range of processes: from manual work processes to software development and typical claims processes. Examples for such processes were the release process of a software tool and the handling of malfunctions of equipment such as the heating or electric sockets within the academic institution. Each process model was built collaboratively during modeling workshops. The goals of these workshops were: (a) to understand the current state of each process and (b) to identify means to improve their efficiency. The process models were not directly related to each other. Each process model was finalized over a series of workshops, varying from 2 to 11 workshops. Overall, 166 workshops were conducted and facilitated by two of the authors of this paper. The facilitator and the participants jointly decided on when a process model was complete. The workshops lasted from 30 minutes to three hours with the facilitator serving as a process analyst. The size of the groups that participated in the workshops varied from 2 to 13 participants which, in this context, are considered as small to medium-size groups. The participants were stakeholders of the respective process that was modeled and they were either directly involved in the process or affected by it. Due to the nature of the processes, the stakeholders came from various backgrounds and domains, from technical personnel to software engineers and administration. None of them had prior experience with using a process modeling notation or modeling processes. The workshops were conducted using the method of the socio-technical walkthrough [21].

The socio-technical walkthrough is based on the assumption that it is less likely to neglect important aspects of a process if it is deliberately inspected step-by-step by a group of stakeholders. During the walkthrough, the facilitator asks the participants predefined questions such as: "*When do you do this task? Who else is involved? What do you need in order to fulfil that task?*". The contributions of the participants are captured and translated into elements of the process modeling notation by the facilitator. In order to keep track of the contributions, an additional person operates a modeling tool. This person integrates the collaboratively proposed modifications into the process model which is continuously presented to the participants, for example using a projector. This allows participants to directly comprehend the effect of their contribution on the process model and ask for revision, if necessary. Throughout these workshops we used the SeeMe process modeling notation and graphical tool [20]. The SeeMe graphical modeling tool records the outcome of the modeling activity (the process model itself) into detailed logfiles that can be analyzed. These logfiles served as a basis for the following analysis.

Methodology

Our research hypothesis is that the evolution of a process model can indicate the maturity and established consensus of the collaborative practice. Subsequently, these indications can inform the facilitators and participants whether it is necessary or reasonable to conduct more workshops or when it is time to conclude. To explore the research hypothesis, we followed a three-step approach:

(1) We analyzed the logfiles collected from the application with respect to participants' activity. We used artifact-related metrics, such as the number of graphical elements for activities, for actors, or for resources. We refer to those as "model metrics".

(2) We transformed the representation of the process models into series of weighted directed networks, one for each workshop, to capture the dynamic nature of process models.

(3) We analyzed and compared the object-oriented and the network oriented metrics on the level of the process models over time. Our purpose was to explore the contribution of each kind of metrics in capturing aspects of the collaborative modeling process.

From Models to Networks: Transformation of Process Models to Directed Graphs

Next, we transformed the process models into directed, dynamic networks. The elements of the process models were represented as graph nodes of different types and the relations between the elements of the process models were represented as graph edges. We used edge weights to capture additional properties of the process models in the network representation. In particular, we followed these transformation rules:

- **Rule 1:** An element of the process model – regardless its type, i.e. resource, actor or activity – is represented in the graph as a node;
- **Rule 2:** A relation of the process model is represented in the graph as an edge;
- **Rule 3:** If one or more elements consist of sub-elements ("embeddings"), then any potential edge that connects this element to the rest of the network has a weight proportional to its sub-elements. For example, if a relation between two elements A and B exist in the process model, and A has x sub-elements and B has y sub-elements, then the resulting edge between A and B in the network representation has an assigned weight of $x * y$, since x sub-elements are related to y sub-elements. The rationale behind Rule 3 is that the weight of a relation should represent all potential connections, either explicit (i.e. between two elements) or implicit (i.e. between sub-elements that are part of the connected element). If a relation originates or points to a sub-element directly, the weight of the resulting edge in the network representation is modified accordingly. An example of this relationship is portrayed in Figure 2.

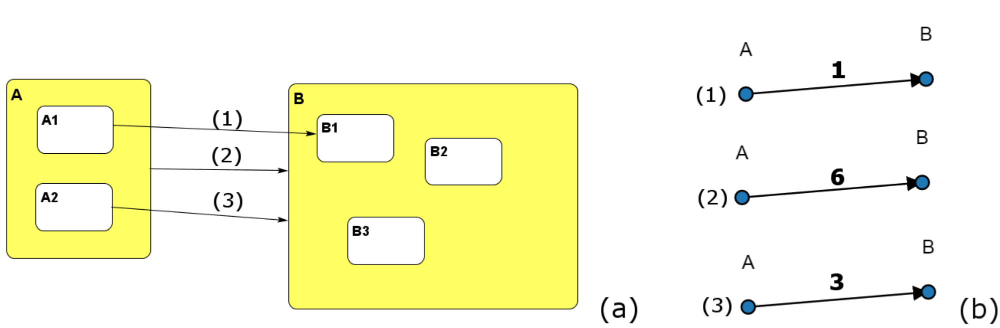

Figure 2. (a) Example of possible embedding relationships between two main elements, A and B and their sub-elements. Element A contains 2 sub-elements (x=2) (A1 and A2), element B contains 3 sub- elements (y=3) (B1, B2 and B3) and (b) the resulting edge weights

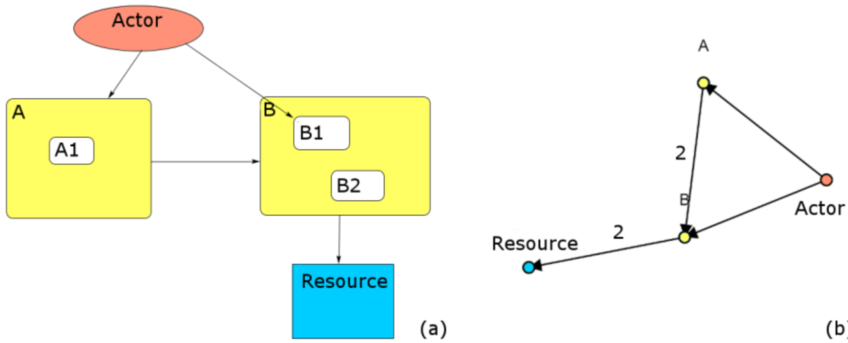

Figure 3. Two representations of the same structure: (a) a diagrammatic representation of a process model created in SeeMe and (b) the corresponding representation of the same process model as a directed, weighted network

This simplified structure allows for exploring the quantitative evolution of the possibly complex process model by applying simple network metrics that might already be sufficient to identify different phases of the progress of modeling. Nonetheless, this is an important aspect of process modeling that we plan to investigate in future work. Figure 3 depicts the diagrammatic representation of a process model and the corresponding representation as a network.

Metrics

For this paper, we calculated different types of process model and network metrics (Table 1). Metrics such as the number of Activities, Actors, Resources and Embedding Relations as well number of nodes (#*Nodes*), the number of edges (#Edges), and the sum of edge weights (*EdgeWeights*) are used as simple descriptive data for structural characteristics of the process model and the network respectively and consequently the level of refinement of the process model. The density is the fraction of existing and possible edges. Path related measures such as the average path length (*APL*) and the diameter (*D*) of the graph can be used to assess the amount of new information contributed by a modification in the network. The diameter of a graph is the greatest geodesic distance between any pair of nodes (incorporating edge weights) where geodesic distance refers to the length of the shortest path between the nodes. Since we cannot assume that the network is strongly connected, i.e. there exists a path between any pair of nodes, the diameter is calculated as the greatest distance only between reachable pairs of nodes. Furthermore, disconnected components often exist in the network. In these cases, the diameter refers to the average diameter of all weakly connected components. If the process model is modified by introducing new relations or by removing existing relations, the diameter of the network can only change significantly if the modification introduces a shortcut between two already connected elements, removed important relations or connects two formerly separated components of the network. If this is not the case, the distances between the elements in the process model are not affected, and thus, the modification cannot be considered as major revision of the process model structure. The same assumption applies for the average path length.

Another type of network measures that can be used to exemplify structural characteristics of the underlying process model is the structural complexity of the network. Changes in the structural complexity of the network representation can result from major changes in the process model. To assess structural complexity, we used the metrics of number of components (#*Components*), components complexity and degree complexity. The number of components refers to the number of independent tasks, i.e. tasks of the process models that are not linked to others which results in disconnected components in the network representation. Note that the direction of edges is ignored in this case (weakly connected components). Complexity of network structures can also be measured in terms of the amount of information the structure contains or the uncertainty to characterize (encode) nodes or other elements of the networks given certain structural properties measured as graph entropy. Various definitions of the entropy of networks exist [14]. The components complexity (*cpx_comp*) corresponds to the entropy of node distributions among components. The greater the total number of components and the more uniformly the sizes of the components are distributed (no typical component size exists), the higher is the complexity of the process model, i.e. typical component size. This clearly distinguishes process models which are heterogeneous in terms of their independent task from process models with many similar structured independent tasks.

Table 1. Description of the main metrics introduced and used in the presented study. The model metrics refer to the metrics that can be extracted directly from the logfiles of the software application while the network metrics refer to the metrics that require the transformation of the process model into a directed network

Model Metrics	**Metrics that refer to the structure of the process model**
#Activities	The number of elements contained in the process model tagged as "Activities"
#Actors	The number of elements contained in the process model tagged as "Actors"
#Resources	The number of elements contained in the process model tagged as "Resources"
#Embedding Relations	The number of inherent relations between elements
#workshops	The number of workshops needed for the completion of a process model
#participants	The average number of participants per workshop
Network Metrics	**Metrics that refer to the structure of the process model expressed as network**
#Nodes	The number of nodes of a network
#Edges	The number of edges of a network
EdgeWeights	The sum of edge weights of a network
Diameter (D)	The average diameter of the graph considering edge weights
Avg_Path_Length (APL)	The average distance between reachable nodes.
Density_Directed	Density of the graph under consideration of directed edges.
#Components (comp)	Number of weakly connected components. Corresponds to the number of independent tasks.
Components Complexity (cpx_comp)	Complexity of the process model regarding disconnected components (independent tasks)
Degree Complexity (cpx_deg)	Complexity of the process model in terms of node connectivity/reachability.

The degree complexity (*cpx_deg*) refers to the complexity of the process model in terms of node degree. The graph is treated as undirected for this measure. It can be interpreted as the uncertainty left about the degree of the nodes. In process models with a high degree complexity the degrees of the nodes are evenly distributed. Thus, those process models cannot be characterized in terms of a typical number of connections the nodes have. In linear structured process models the nodes would have a typical low degree while in more complex process models there are nodes with a wide range of degrees. If new nodes do not change the structure of the process model significantly, the uncertainty about the degree of a node does not change significantly either. Additionally, the number of workshops and the number of participants per workshop were also taken into account in the analysis.

ANALYSIS AND RESULTS

Descriptive Analysis

On average, the process models were created in 1.6 hours from teams of roughly 4 participants over 6 workshops and they consisted of about 75 elements in total. Figure 4, depicts the evolution of a process model from the first workshop to the final project. In particular, we present the process model after the end of the first, the third, the sixth and the final (tenth) workshop, as well as the representation of the process model as a network.

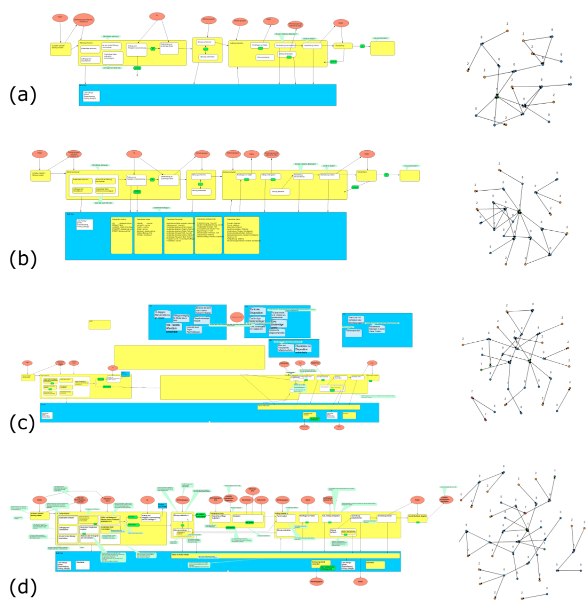

Figure 4. Evolution of a process model over time: the process model and the resulting network (a) after the first workshop, (b) after the end of the third workshop, (c) after the end of the sixth workshop and (d) after the end of the final (tenth) workshop. A higher-quality figure is provided in the Appendix (Figure 10).

Analysis of the Metrics Over Time

To explore the evolution of process models, we compared how the model and network metrics (Table 1) changed over time. We computed the difference of the various metrics' values between consecutive workshops. Then, we calculated for which consecutive workshops these differences were maximized. Figure 5 presents the sequence of workshops in which major changes took place. The x-axis represents the metrics used in this study. The bars represent the sequence of consecutive workshops (i.e. 1->2 the first to the second workshop). The y-axis represents the number of process models with most differences. The analysis showed that major changes took place between the first to second workshops or between the second to third workshops. Changes decrease as the number of workshops increases. This finding may suggest that the main modeling process unfolds during the first two to three workshops, in particular between the second and the third workshop. This may suggest the need for support during the early phases since the facilitator can decide early on whether the result is satisfying in terms of maturity and completeness to avoid unnecessary extra effort.

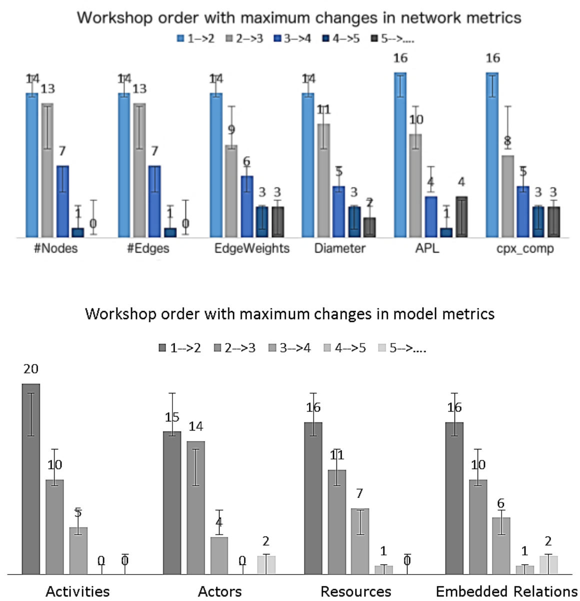

Figure 5. Workshops in which major changes took place (based on calculated differences of metrics' values between consecutive workshops). The x-axis represents the model and network metrics used in this study, the bars represent the consecutive workshops (i.e. 1->2 the first to the second workshop) and the y-axis represents the number of process models that appeared to have the maximum difference for this sequence of workshops

Correlation Analysis

We conducted a correlation analysis (Spearman's correlation) in order to explore potential relations between workshop characteristics (such as the number of participants or the number of workshops) and the model/network metrics. We correlated the number of workshops (#*workshops*) and the number of workshop participants (#*participants*) with the metrics of Table 1. Table 2 presents the statistically significant results of the correlation analysis.

Table 2. Correlations between number of participants and number of workshops and network metrics. The correlations marked with double asterisk (**) are statistically significant on the p<.01 level. The correlations marked with single asterisk (*) are statistically significant on the p<.05 level.

	Diameter	APL	Components Complexity
#participants	.110	.339*	.491**
#workshops	.201	.355*	-.206

On one hand, the number of elements, such as Activities, Actors and Resources and the related network metrics, such as the number of nodes and the number of edges, did not correlate to the number of workshops or to the number of participants. One could argue that the longer the modeling process lasts (i.e. the more workshops are conducted) and/or the more people get involved, the more information would come into the process model leading to increased volume of activity and elements. However, the data did not confirm this hypothesis. On the other hand, the number of participants correlated positively with network metrics, such as the average path length and the components complexity. The contribution of the participants is not depicted as increased number of elements but it is expressed as changes in the structure of the network. This can also suggest information diversity since the participants of workshops usually come from different backgrounds. Furthermore, the number of workshops correlated positively with the Average Path Length (*APL*). This may suggest that even though there is no new information input after a number of workshops, there are additional refinements in the structure of the process model which might affect its overall quality. Metrics such as the average diameter of the disconnected components or the density of the network do not correlate either to the number of workshops or to the number of participants. Other metrics, such as the average duration of workshops, were also studied without statistically significant results.

EVALUATION

As previously mentioned, we identified that the main modeling activity takes place roughly during the first three workshops. This means that not much content is added afterwards and only minor changes are conducted. These minor changes however can still be important. In order to assess what kind of changes happen before and after the third workshop, we conducted a supplementary qualitative analysis. For this analysis, we asked two experts to code process models individually. For each process, we conducted two comparative codings: in the first we compared the process model after the second workshop with the process model after the third workshop and in the second we compared the process model after the third workshop with the last process model. Codes were applied to phases of a process in order to arrive at a finer grained coding than just applying codes to the process model as a whole. The phases were selected by the two process modeling experts and agreed upon before the coding started. For the coding, we assumed the last process model as complete since the modeling process had ended there and consensus among all workshop participants was reached.

Table 3. Significant correlations between codes.

	C1	C3	C4	C5	C7
C4	0.60***	0.27**			
C7			0.36***		
C8	0.47***	0.25*	0.53***		0.27*
C9				0.29**	

We developed a coding scheme (Table 5 in the appendix) that focusses on relevant aspects of a process and that covers types of modifications to process models. It covers major modifications such as the creation of a new phase (C0), the addition of new process steps, actors and resources (C1) as well as their removal (C2). The coding scheme also includes more detailed aspects of a process model such as the modification of sequences (C3a to C3c), the addition of details to existing process steps, actors or resources (C4), the alteration of relations between actors and process steps or between process steps and resources (C5) as well as the modification of conditions (C6). Finally, our coding scheme also covers minor modifications to a process model such as changing the descriptions of elements (C7a and C7b), the addition of textual annotations (C8) and aesthetic changes to the overall appearance of a process model (C9) which do not alter the semantics of the process model. We asked the coders to apply codes only once per phase (e.g. C1 was applied when one or multiple elements had been added to a phase) since we were focusing on the quality of changes to a process model rather than the quantity of such since this aspect was already part of the previous evaluation. We refined the initial coding scheme through multiple iterations with two versions of process models that are not part of the dataset. With respect to the analysis we expected codes 0 through 3 to be mainly applied for the comparison between the process model after the second and after the third workshop since these codes represent major changes to a process model while the latter codes represent minor changes. These are consequently expected to be applied more for the comparison between the process model after the third workshop and the last process model.

We started our analysis by assessing the inter coder agreement through Cohens-Kappa [13] (Table 6 in the appendix). Following the guidelines by Landis and Koch [28] we found moderate (0.41 – 0.60) to substantial (0.61 – 0.80) scores for all codes except for codes 3c (0.325) and 7b (0.317). We thus combined codes 3a to 3c to a single code 3 and codes 7a and 7b to a single code 7. This resulted in overall moderate to substantial scores. We also conducted a correlation analysis in order to assess whether two codes capture the same aspect of changes to a process model. We found codes 1 and 4 to be strongly and significantly correlated. This correlation however can be expected since adding details to a process and adding elements to a process oftentimes represent the same thing. The other codes either did not correlate or were only weakly correlated. The significant correlations are depicted in Table 3.

After this initial analysis, we compared the development of the codes between workshop 2 and 3 and between workshop 3 and the last one by computing the relative standard deviation (RSD) for each code. Figure 6 presents the results.

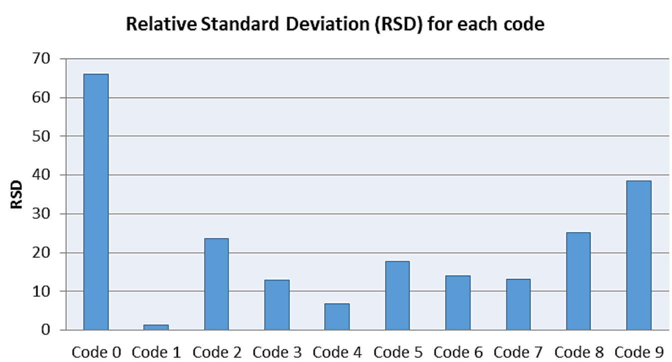

Figure 6. The relative standard deviation of the weighted averages of the codes between WS2 and 3 and WS3 and the last WS.

The results confirm our prior analysis; i.e. major changes to the process models happened between the second and third workshop. This is evident by the high RSD with respect to adding new phases (RSD_C0 = 66%), removing elements or moving them to different phases (RSD_C2 = 23.71%) and relation between process steps (RSD_C3 = 13%) and other elements (RSD_C5 = 17.71%). The changes that happened between workshop 3 and the last one were mainly changes to conditions (C6), changes to the descriptions of elements (C7), annotations (C8) and aesthetic changes (C9). These changes bear only minimal conceptual significance since aesthetical changes as well as annotations do not influence the semantics of a process model. Also, the changes that were conducted to the descriptions of elements only focused on making them more expressive or making sure that all elements that represent the same are also labeled the same. Finally, the changes to conditions indeed alter the semantics of the process model but they are usually solely reflected by the number of cases that are conducted within one or the other sequence of actions (backwards facing arrows in Figure 1). They can thus also be qualified as minor changes to a process model.

These results indicate that changes after the third workshop focused on details rather than on adding additional content. This conforms to typical evolution of collaboratively created process models [3]. It however provides an indication that the quantitative metrics described previously are indeed suitable to capture the evolution of process models and could subsequently be used for continuous feedback to facilitators and participants.

FROM METRICS TO PREDICTION

In order to support facilitators and participants of workshops, it is necessary to move from the exploration of metrics to predictions with respect to the evolution of process models. In this section, we describe a computational approach to move from metrics to predictions. It should however be noted that the objective of this section is not to propose a predictive model but to illustrate how analytics can support facilitators.

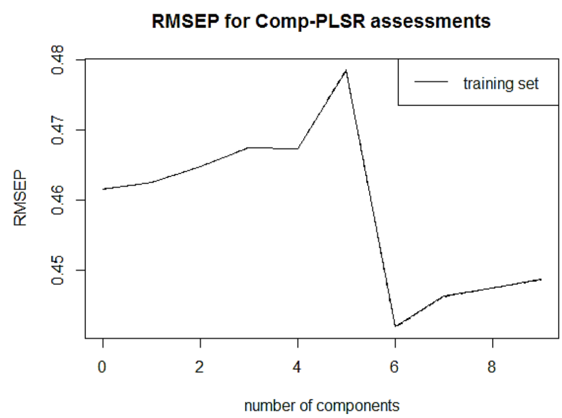

Figure 7. The Root Mean Square Error of Prediction (RMSEP) of the Comp-PLSR model during the training phase with respect to the number of components.

We have implemented a partial least squares regression (PLSR) predictive model (we refer to it as Comp-PLSR) to predict the evolution of process models in terms of their completeness [9]. We used partial least square regression as a modeling method in order to avoid multicollinearity issues between predictive features. The predictive model takes as input (predictive features) a set of 9 features, as presented in section *Method of the study*. These features are: the workshop number, the number of participants, the number of nodes, the number of edges, the sum of edge weights, the number of components, the average path length, the diameter, the number of weakly connected components and the complexity of a process model regarding disconnected components. Based on these features, the Comp-PLSR model provides an assessment of the process model's completeness in a range from 0 to 1 where 1 indicates a complete process model.

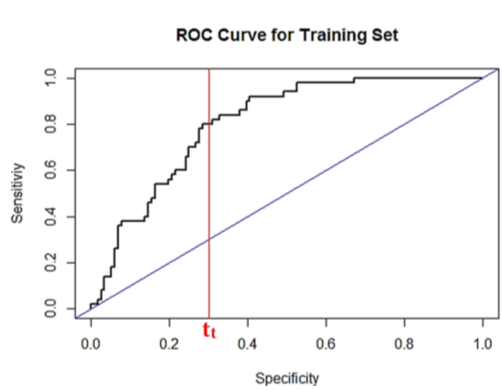

Figure 8. ROC Curve for the Comp-PLSR binary classifier and optimal threshold t_t

We used the original dataset of 35 process models as our training dataset. The completeness of the process models of the training dataset was assessed based on the outcome of the coding process. We considered process models that had minor modifications after the third workshop (codes 6 to 9 in Table 5 in the appendix) as complete. The predictive model was fitted model with 9 components and including leave-one-out (LOO) cross-validated predictions. Figure 7 depicts the Root Mean Square Error of Prediction (RMSEP). It shows that we need 6 components to minimize the error.

We assess the completeness of a process model as a binary variable (0 for incomplete process models, 1 for complete process models). Therefore, the predictive model is practically a binary classifier and we should determine the optimal classification threshold t_{t}. To that end, we use the receiver operating characteristic (ROC) curve of the classifier and identify the optimal threshold that minimizes the error frequencies (false negatives). In this case, the optimal threshold is t_t= .301 and it is depicted in Figure 8. The confusion matrix is presented in Table 4.

Table 4. Confusion matrix for the predictive model Comp-PLSR used as binary classifier to assess process models' completeness, where Class 0 indicates an incomplete process model and Class 1 indicates a complete process model. The accuracy of the binary classifier as computed from the confusion matrix was Acc= 0.638.

Reference / Prediction	0	1
0	74	42
1	18	32

As a test set, we used four process models that were created in the same way and within the same context with the process models of the training set, but they were left out the training set.

The assessments of the Comp-PLSR model for the process models of the test set are depicted in Figure 9. The dotted lines depict the optimal threshold for classifying process models as complete. Figure 9 indicates that 3 out of 4 process models (process models A, B and D) reached this threshold around workshop 3 and 4. Only process model C never reached the threshold as the modeling process stopped before (on workshop 3). The assessments of process model completeness, as predicted by the Comp-PLSR model, correlates to the number of modeling workshops. That means, the more workshops a group devotes for the construction of a process model, the more complete the process model will be. This relation between number of workshops and completeness is not surprising since new workshops potentially lead to changes and refinements. However, facilitators and participants should compare the benefits gained from these refinements with the cost of organizing additional workshops (participants' effort, time requirements, required resources, etc.). For example, it is possible that the facilitator and participants of the modeling workshops for process model D (Figure 9) chooses to stop the modeling workshops after the end of the fifth workshop since by then the prediction probability that the process model is complete is roughly 60%. Further refinements might not have a big impact on the purpose of the process model and the cost of bringing the participants together is too high. On the contrary, the facilitator and participants who are part of the workshops for process model C (Figure 9) may want to extend the modeling workshops instead of stopping after the third workshop where the prediction is that the process model is incomplete.

In order for such an approach to be useful, we should first ensure the validity and accuracy of the prediction model as well as the appropriateness of the modeling method. We want to point out that the purpose here is only to describe how such an approach could be used to support facilitators and participants and to scaffold modeling workshops. The size of the dataset does not allow for rigorous validation. However, the implementation process of a predictive model and its validation is part of future work. The features-list used as predictors should be further refined, studied and potentially expanded. Furthermore, we also have assessed completeness as a binary variable. However, it may be meaningful to assess completeness on a value range. This will allow us to provide feedback with respect to how much more effort and time might be necessary to provide a complete process model.

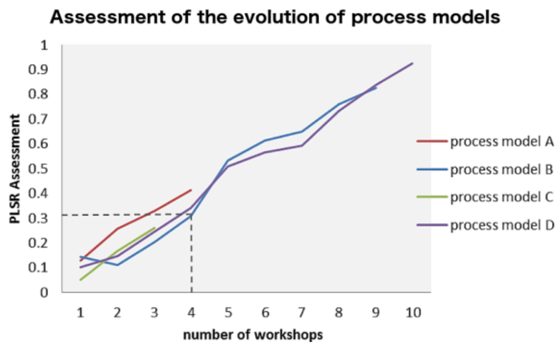

Figure 9. PLSR Assessment of the evolution of process models with respect to completeness for 4 process models constructed collaboratively over multiple workshops. The optimal threshold for considering a process model as complete is PLSE Assessment =.301.

DISCUSSION

In this paper, we introduced network metrics – commonly used in Social Network Analysis (SNA) – to assess the evolution of collaboratively-created process models. We explored the connection between the evolution of such collaboratively created artifacts and the collaborative process of creating them. In particular, we focused on identifying when a process model has reached a satisfactory level of completeness using computational methods. Second, we proposed a computational approach to provide feedback and support to facilitators and participants during modeling workshops.

Results indicate that the evolution of process models can indeed serve as a basis for assessing collaborative practice. They also indicate that the number of workshops needed for the completion of a process model does not correlate with the number of the elements or the size of the process model. Process models that took three or more workshops to be completed resulted in more elaborate and detailed representations – but not necessarily bigger in size – than the process models that were completed in fewer workshops. The results also suggested that the design process required, roughly, three workshops to provide a complete process model. In later workshops, participants refined the process models but there were no further significant changes in the structure or the size of the process models as indicated by the qualitative results reported in the previous section. This finding suggests that the third workshop signifies a decision-making point: stakeholders should clarify what has been done so far, what still needs to be done and which important matter still has to be discussed and decided on. For facilitators, this might require adapting their practices in order to achieve the desired outcome in an efficient way. For example, after the third workshop facilitators should clarify which important decisions still have to be taken before it is possible to test a process in practice rather than attempting to discuss every aspect of a process theoretically by conducting additional workshops.

The computational model proposed in this work can support the stakeholders in directing effort where necessary by assessing the state of a process model and predicting when the model will reach a sufficient level of maturity. Based on the feedback provided by the computational model, the stakeholders can still continue workshops if desired to refine further details or work on other necessary processes.

Implications For Theory and Practice

This study allowed us to gain deeper insight into how process models evolve when they are created collaboratively. The results confirmed that major changes to process models happened during the first three workshops. Later changes mainly focused on addressing details and making refinements that can – but need not – lead to important decisions. Thus, it can be stated that more workshops do not necessarily lead to better results in terms of process model completeness and do not justify the additional overhead. The study also revealed that the number of participants affects volume and diversity of information depicted in the process model since each participant has an individual perspective on the process. This does not have an effect on the size of a process model – that is "more participants" does not necessarily mean more elements – but it may be an indication of richness and diversity with respect to the information contained. Our proposed approach can help facilitators and participants to design, plan and run collaborative modeling workshops in an efficient way while producing high quality process models. To that end, this approach could support planning and decision-making on three levels: a) to provide assessments of the current state of a process model, b) to provide recommendations for potential modifications (for example, bringing in people from different backgrounds to participate in the modeling workshops or minimizing the number of participants when we have ensured that different aspects have been covered) and c) to provide information that aim to help facilitators and participants to decide whether or not it is useful to conduct future workshops.

Another implication of our work is that we established the applicability of SNA in the context of collaborative process modeling. At this, transforming a process model to a network yields a higher level of abstraction that still preserves the main structural features that can be captured by established metrics rooted in graph theory. Specifically we found indications for the feasibility to use SNA to assess the state of process models and to provide insight and feedback to workshop participants and facilitators. This feedback can support workshop facilitators in making decisions and it can simplify and augment communication between facilitators and stakeholders. Even though we used a specific process modeling notation for the construction of the process models, it can be argued that the results are applicable to other process modeling notations since all notations are equivalent in their composition since they all provide symbols for the basic constructs of processes: activities, actors, resources, decisions and flow. The results may also extend to other collaborative practices in which graphical visualizations are jointly created such as concept and decision maps [2,43] or software architecture [40]. SNA methods have a wide range of application areas but – to the best of our knowledge – they are not widely adopted to support collaborative practices in business process modeling contexts. We believe that this research may be interesting for SNA researchers and can serve as an application field for designing and testing new or existing algorithms and tools.

Limitations

This work has limitations with respect to the methodology and the analysis. First, we conducted the study in one organization, which poses a threat to the generalizability of our results. It should be noted that both the participants of the workshops as well as the processes that we analyzed covered a large variety of different domains and contexts which mitigates this effect. In addition, we perceive this work to be exploratory in nature and as such it is meant to rather serve as a proof of concept for the application of SNA in the context of co-located collaboration. We also argue that this work provides an insight into how the evolution of collaboratively created artifacts can serve as basis to assess the state of collaboration as well as a starting point for future work. Another limitation of our work is the focus of analyzing the state of process models after each workshop. It can be argued that continuously analyzing the evolution of process models during workshops might be more meaningful with respect to assessing collaboration. Assessing the state of process models during a workshop however would have posed the threat that we consider a state that has not been agreed upon by all participants since changes to a process model might and will be reverted during a workshop while the end-result of a workshop can be perceived as a consensus – or as an appropriate compromise – between all participants. Finally, our research builds on the assumption that consensus has been reached after each workshop. While this assumption is true on the basis that participants vocally agreed on the state of a process model at the end of a workshop, this agreement might be revoked e.g. after carefully considering its impact. In addition, consensus should not be confused with correctness or usefulness. Participants can certainly agree on a process model that proves not be useful at the end and they can as well be wrong about their assumptions about how a process is or will be executed in the real world. We also expect that the evolution of process models is highly depending on factors such as the complexity and size of the processes that are modeled, the experience of the facilitator etc., that are not considered in this study.

CONCLUSION AND FUTURE WORK

This study employs SNA methods to automatically analyze the evolution of collaboratively created artifacts in order to assess the current state of collaborative practice. Our results indicate the feasibility of such analytics and they also provide hints towards their application as a feedback mechanism for both participants and facilitators of collaborative modeling workshops. We also provide an in-depth analysis into the evolution of collaboratively created process models.

In future work, we plan to study how we can include information of semantic and syntactic incompleteness to further assess the maturity of process models. Furthermore, we expect that the evolution of process models highly depends on factors such as the complexity and size of the processes that are modeled, the experience of the facilitator etc., that are not considered in this study. We plan to investigate these parameters and their effect on collaborative modeling, as well as to study how we can use information of semantic and syntactic incompleteness to further assess the maturity of process models. Furthermore, we plan to define a general framework and to provide rubrics for the assessment of the evolution of collaboration and the semi-automated evaluation of its outcome.

ACKNOWLEDGEMENTS

We would like to thank Prof. H. Ulrich Hoppe and Christopher Lentzsch for their valuable contribution.

References

[1] Mark S Ackerman, Juri Dachtera, Volkmar Pipek, and Volker Wulf. 2013. Sharing knowledge and expertise: The CSCW view of knowledge management. *Computer Supported Cooperative Work (CSCW)* 22, 4–6: 531–573.

[2] David F Andersen, Jac AM Vennix, George P Richardson, and Etiënne AJA Rouwette. 2007. Group model building: problem structuring, policy simulation and decision support. *Journal of the Operational Research Society* 58, 5: 691–694.

[3] D.F. Andersen and G.P. Richardson. 1997. Scripts for group model building. *System Dynamics Review* 13, 2: 107–129.

[4] Christopher Andrews, Alex Endert, and Chris North. 2010. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 55–64.

[5] Wasana Bandara, Guy G. Gable, and Michael Rosemann. 2005. Factors and measures of business process modelling: model building through a multiple case study. *European Journal of Information Systems* 14, 4: 347–360.

[6] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439: 509–512.

[7] R.P. Bostrom, R. Anson, and V.K. Clawson. 1993. Group facilitation and group support systems. In *Group support systems: New perspectives*, L.M. Jessup and J.S. Valacich (eds.). Macmillan, 146–168.

[8] Robert Briggs and Gert-Jan de Vreede. 2009. *ThinkLets: Building Blocks for concerted Collaboration*. Center for Collaboration Science.

[9] Bo Cheng and Xizhi Wu. 2006. A modified PLSR method in prediction. *J. Data Science* 4: 257–274.

[10] Irene-Angelica Chounta and Nikolaos Avouris. 2014. Towards the real-time evaluation of collaborative activities: Integration of an automatic rater of collaboration quality in the classroom from the teacher’s perspective. *Education and Information Technologies*: 1–21.

[11] Irene-Angelica Chounta, Tobias Hecking, Heinz Ulrich Hoppe, and Nikolaos Avouris. 2014. Two Make a Network: Using Graphs to Assess the Quality of Collaboration of Dyads. In *Collaboration and Technology*. Springer, 53–66.

[12] Herbert Clark and Susan E. Brennan. 1991. Grounding in Communication. In *Perspectives on Socially Shared Cognition*, Lauren B. Resnick, John M. Levine and Stephane D. Teasley (eds.). American Psychological Association, 127–149.

[13] Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin* 70, 4: 213.

[14] Matthias Dehmer and Abbe Mowshowitz. 2011. A history of graph entropy measures. *Information Sciences* 181, 1: 57–78.

[15] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A Reijers. 2013. *Fundamentals of Business Process Management*. Springer.

[16] Erik Duval. 2011. Attention please!: learning analytics for visualization and recommendation. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, 9–17.

[17] Paul Erdos and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5, 1: 17–60.

[18] Paul Harmon and Celia Wolf. 2016. The state of business process management. *Business process trends*.

[19] Marielle den Hengst. 2005. Collaborative Modeling of Processes: What Facilitation Support Does a Group Need? In *AMCIS 2005 Proceedings*, 73–80.

[20] Thomas Herrmann. 2006. SeeMe in a nutshell - the semi-structured, socio-technical modeling method.

[21] Thomas Herrmann. 2009. Systems design with the socio-technical walkthrough. *Handbook of research on socio-technical design and social networking systems*: 336–351.

[22] Thomas Herrmann, Gabriele Kunau, and Kai-Uwe Loser. 2007. Socio-technical self-description as a means for projects of introducing computer supported cooperation. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, 232c–232c.

[23] Thomas Herrmann and Alexander Nolte. 2014. Combining Collaborative Modeling with Collaborative Creativity for Process Design. In *COOP 2014 - Proceedings of the 11th International Conference on the Design of Cooperative Systems, 27-30 May 2014, Nice (France)*, 377–392.

[24] Thomas Herrmann, Alexander Nolte, and Michael Prilla. 2013. Awareness support for combining individual and collaborative process design in co-located meetings. *Computer Supported Cooperative Work (CSCW)* 22, 2: 241–270.

[25] H. U. Hoppe, J. Engler, and S. Weinbrenner. 2012. The impact of structural characteristics of concept maps on automatic quality measurement. In *International Conference of the Learning Sciences (ICLS 2012), Sydney, Australia.*

[26] Gwendolyn L Kolfschoten, Robert O Briggs, Gert-Jan De Vreede, Peter HM Jacobs, and Jaco H Appelman. 2006. A conceptual foundation of the thinkLet concept for Collaboration Engineering. *International Journal of Human-Computer Studies* 64, 7: 611–621.
[27] John Krogstie. 2012. *Model-based development and evolution of information systems: A Quality Approach*. Springer Science & Business Media.
[28] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*: 159–174.
[29] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1: 2.
[30] Roberto Martinez, Judy Kay, James R. Wallace, and Kalina Yacef. 2011. Modelling symmetry of activity as an indicator of collocated group collaboration. In *User Modeling, Adaption and Personalization*. Springer, 207–218.
[31] Jan Mendling. 2010. Foundations of Business Process Modeling. In *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, M.R. Syed and S.N. Syed (eds.). IGI Global, 189–222.
[32] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. 2007. What makes process models understandable? In *Business Process Management*. Springer, 48–63.
[33] OMG. 2003. *UML 2.0 Superstructure - Final Adopted Specification*. Object Management Group.
[34] OMG. 2006. *Business Process Modeling Notation (BPMN) Specification*. Object Management Group.
[35] Michael Prilla, Alexander Nolte, Thomas Herrmann, Gwendolyn Kolfschoten, and Stephan Lukosch. 2013. Collaborative Usage and Development of Models: State of the Art, Challenges and Opportunities. *International Journal of e-Collaboration (IJeC)* 9, 4: 1–16.
[36] Jan Recker. 2010. Opportunities and constraints: the current struggle with BPMN. *Business Process Management Journal* 16, 1: 181–201.
[37] Peter Rittgen. 2010. Collaborative Modeling: Roles, Activities and Team Organization. *International Journal of Information System Modeling and Design (IJISMD)* 1, 3: 1–19.
[38] Peter Rittgen. 2013. Group consensus in business process modeling: A measure and its application. *International Journal of e-Collaboration (IJeC)* 9, 4: 17–31.
[39] E.A.J.A. Rouwette, J.A.M. Vennix, and T. Mullekom. 2002. Group model building effectiveness: a review of assessment studies. *System Dynamics Review* 18, 1: 5–45.
[40] Richard Soley and others. 2000. Model driven architecture. *OMG white paper* 308, 308: 5.
[41] Amy Soller, Alejandra Martínez Monés, Patrick Jermann, and Martin Muehlenbrock. 2001. From mirroring to guiding: a review of state of the art technology for supporting collaborative learning. In *Proceedings of the First European Conference on Computer-Supported Collaborative Learning*.
[42] Denis Ssebuggwawo, Stijn Hoppenbrouwers, and Erik Proper. 2010. Assessing collaborative modeling quality based on modeling artifacts. In *The Practice of Enterprise Modeling*. Springer, 76–90.
[43] William MK Trochim. 1989. An introduction to concept mapping for planning and evaluation. *Evaluation and program planning* 12, 1: 1–16.
[44] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of ‘small-world’networks. *nature* 393, 6684: 440–442.

Received April 2017; revised July 2017; accepted November 2017.

Appendix

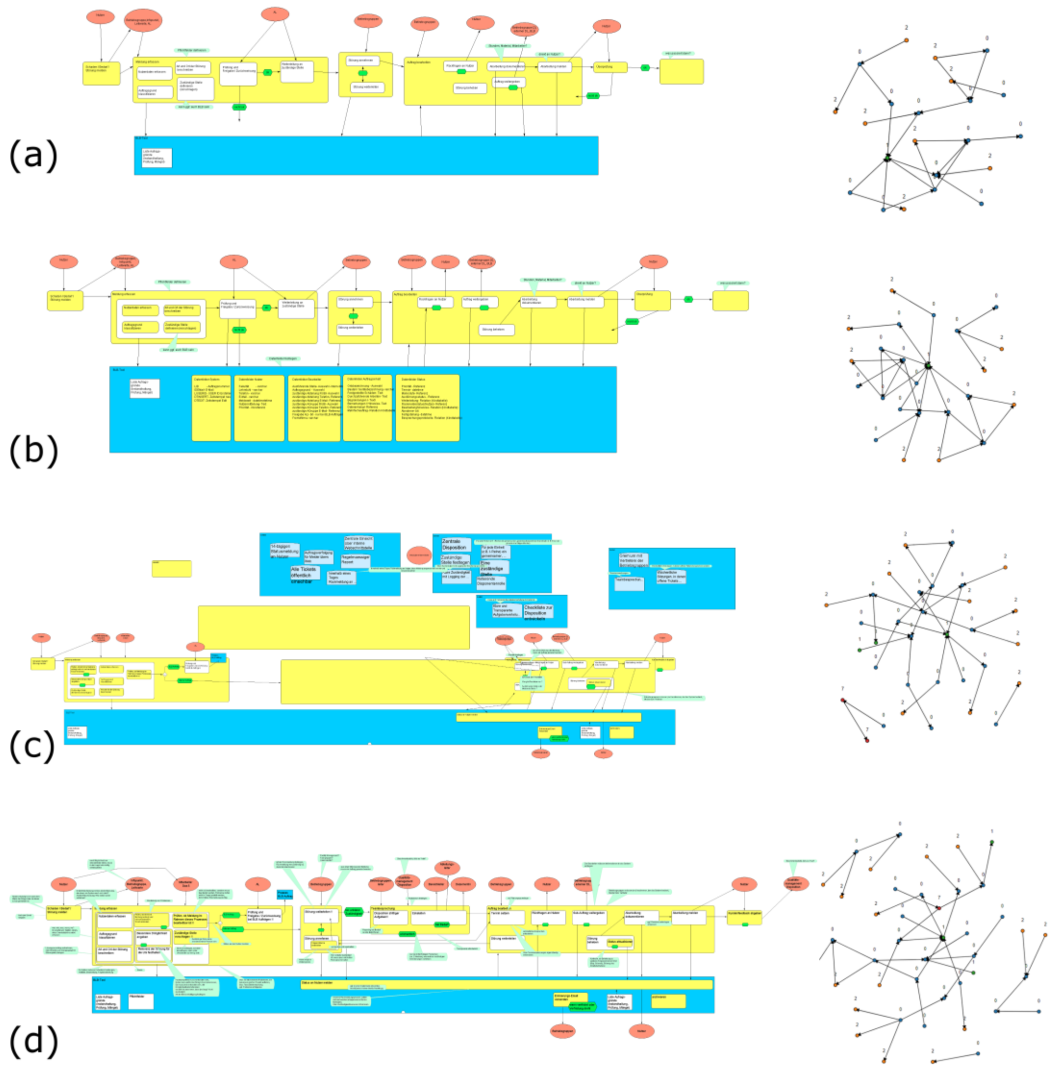

Figure 10. Evolution of a process model over time: the process model and the resulting network (a) after the first workshop, (b) after the end of the third workshop, (c) after the end of the sixth workshop and (d) after the end of the final (tenth) workshop.

Table 5. **Coding scheme for changes on process models**

Code	description
C0	A **new phase** was added to a process model (no further codes required)
C1	**New process steps**, **new actors**, **new resources** or **new results** from process steps were added
C2	Process steps were **removed** or **moved to different phase**
C3a	Elements have been put **into a sequence** that were not in a sequence before (only applied if all elements existed before)
C3b	A **sequence** of elements has been **disbanded** (only applied if all elements in that sequence existed before and still exists afterwards)
C3c	Existing sequence has been **altered** within a phase (only applied if all elements in that sequence existed before and still exists afterwards)
C4	**Details were added** to or **removed from** existing process steps, actors, resources or results from process steps
C5	**Relations** between actors and process steps or between process steps and resources and results **were altered** (does not include visual embedding since this is covered by C4)
C6	**Conditions** were added, removed or modified on *existing* process steps, actors, resources, results or relations between such (only applied when element and / or relation existed before and still exists afterwards)
C7a	Elements were renamed to further **specify their meaning**
C7b	Elements were renamed to make the wording more **consistent**
C8	**Annotations** were added, removed or modified (only applied if element that is annotated existed before and still exists afterwards)
C9	Aesthetic changes

Table 6. **Cohen's-Kappa for codes C1 to C9**

C0	C1	C2	C3a	C3b	C3c	C3	C4	C5	C6	C7a	C7b	C7	C8	C9
0.717	0.842	0.591	0.634	0.562	0.325	0.695	0.703	0.578	0.553	0.547	0.317	0.778	0.793	0.608