

MoRoCo 2013: Models and their Role in Collaboration

Alexander Nolte¹, Michael Prilla¹, Peter Rittgen², Stefan Oppl³

¹Department Information and Technology Management, University of Bochum,
Germany

nolte@iaw.rub.de, prilla@iaw.rub.de

²School of Business and IT, University of Borås, Sweden

peter.rittgen@hb.se

³Department of Business Information Systems, Johannes Kepler University of
Linz, Austria

stefan.oppl@jku.at

Abstract. Using visual representations of work or business processes can be considered a common practice in modern organizations. These models serve a large variety of different purposes such as documentation of current practices, or informing and planning change or software development. Given the nature of work and businesses they reflect it is reasonable to develop and use them collaboratively. There are, however, also many downsides to collaborative model usage and development in current practice. Among others, models are often not fully understood and are thus not used by people who work in the processes the models represent, resulting in limited impact of process redesign on everyday work. Furthermore, only a minority of people within organizations actually use models, even though they have been proven to be very useful especially for collaborative work. Given the increasing popularity of models in organizations, understanding and defining their role in collaboration is of vital interest for the CSCW community and therefore this workshop aims at bringing together researchers and practitioners and forming a community for research in this area.

Introduction

The usage of visual representations of static parts of an organization (e.g. diagrams depicting hierarchies in the organization's structure or a company's competences), dynamic aspects (e.g. work and business processes) or results of creative problem-solving sessions (e.g. brainstorming results) can be considered a common practice in modern organizations. These visual representations include process models, conceptual models and mind maps. They are used for multiple tasks such as software development, design and engineering, process optimization and reengineering as well as marketing and strategic development. Obviously, these models are hardly ever artifacts that are used and developed by single users for their own personal needs. They are rather developed for larger target groups throughout an organization to support them in sense making and creating a shared understanding about cooperative work and its interfaces. Consequently, they are both used by many people and developed collaboratively. However, the number of people that are affected by these representations is usually larger than the number of people who participate actively in their development. The need to create communicable and comprehensible models is thus evident.

Alongside the increasing usage and popularity of visual representations in organizations, there also is growing interest in their usage and development in the CSCW¹ community. This comprises not only the usage and development by modeling experts, but explicitly takes users into account that are no experts in modeling, thus including factors that might motivate or hinder them to use models and actively participate in their development. The emerging importance of this new field of CSCW research is reflected by workshops (e.g. "TAProViz" at BPM 2012 and "CollabViz" at ECSCW 2011), tracks at international conferences (e.g. "Collaborative Modeling" at HICSS 2009, 2010, 2011 and 2012), papers at various CSCW related conferences (e.g. Baacke, Rohner, Winter & Fitterer, 2009; Brosch, Seidl, Wieland, Wimmer & Langer, 2009; Herrmann & Nolte, 2010; Klebl, Hackel & Lukosch, 2009; Nolte & Prilla, 2012), journal contributions (Heer, Bostock & Ogievetsky, 2010; Renger, Kolfshoten & De Vreede, 2008; Rittgen, 2010; Yuille & Macdonald, 2010) and journal special issues (Prilla, Nolte, Herrmann, Kolfshoten, & Lukosch, 2013; Rittgen, 2009, 2012). Additionally, there are various parallel approaches in related research communities such as Group Decision Support, Business Process Management and Group Support Systems.

However, despite the fact that modeling is a popular approach in practice and thus, many models exist in organizations, they are only used by a minority of the people. This consequently leads to them only playing a minor role in everyday work of the employees of an organization. This is quite surprising considering the fact that models have proven to be very useful for cooperative work, especially

¹ Computer-Supported Cooperative Work

when planning it. Furthermore, the number of people creating models stands in stark contrast to the number of people that are actually affected by planning based on these models. Even if they are created collaboratively by process stakeholders, they often have little impact on the people that are actually working in these processes (cf. Prilla, 2010) and thus do not transcend into work practice. The reasons for this are manifold. First, there are few insights on how to spread models and sustain their usage in organizations thus coupling them with activities and artifacts of everyday work. This explicitly includes a lack of knowledge about factors that might motivate or hinder model usage and development. Furthermore, up to now, little is known about how people interact with models that are no modeling experts. By interaction of these non-expert users, we not only refer to model creation, but also their usage in people's daily work for e.g. discussion, knowledge elicitation and creating a common understanding. Non-expert interaction with them however proves to be an issue, as people that are involved in processes usually are no modeling experts. Interaction in this context includes enabling people to use modeling languages and thus to directly contribute to model development, as well as providing other means such as textual or visual annotations to enable indirect contributions. This leads to the question of how models can be coupled with other artifacts of everyday work, which might prove to be beneficial for their usage and ultimately increase their impact.

Besides the usage of models by non-experts, there is an additional research gap in the collaborative construction of visual representations. Usually, the creation and modification of models is restricted to collocated workshops and similar modes of interaction and collaboration, where experts are required to facilitate and support the modeling process. Despite their applicability and feasibility in many situations, these workshops simply do not fit the need to rapidly adjust processes to changing conditions inside and outside an organization. Given the distributed nature of many organizations, these workshops also do not sufficiently reflect the need to include expertise distributed across different locations. Therefore, finding ways to enable dislocated users to contribute actively to model creation and maintenance in a collaborative modeling process is necessary.

Given the increasing usage of visual representations in organizations, their collaborative and distributed use, creation and sustainment is of vital interest for the CSCW community, which has a long tradition of researching the usage of common artifacts, the influence on collaboration by artifacts and their collaborative creation. This workshop therefore aims at being a starting point in forming a community for research in this area.

It is a follow up to a workshop on "Collaborative usage and development of models and visualization" which was held at ECSCW 2011 in Aarhus. Proceedings of which can be found online at <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-777/>. Selected papers from the workshop

will also be published in a special issue of the International Journal for eCollaboration (Prilla et al., 2013).

Scope and aim of MoRoCo

The goal of MoRoCo is to bring together researchers, lecturers and practitioners from different fields, who are interested in the collaborative usage and development and maintenance of structured visual representations such as process models, conceptual models or mind maps. This includes experiences from empirical case studies, teaching and the introduction of models and modeling into organizations. The workshop aims at building a large picture of research on the role that models play in collaborative work in order to set up a common research agenda. The topics of the workshop thus include but are not restricted to:

- The process of cooperative modeling: design cycles, model negotiation, view integration, roles of participants in modeling, team organization, etc.
- Sustaining model usage and maintenance in organizations
- Motivating involvement and active usage of models
- Involving non-experts in model development and usage
- Increasing the range of involvement: from core stakeholders to all stakeholders
- Coupling models with activities and entities of work
- Roles of models for collaboration e.g. guides / maps
- Models as instruments for consensus building
- The role of models in spanning inter or intra organizational boundaries
- Integrating visual modeling and model dialogues in natural language
- “Meta”-modeling: structuring the dialogue around models
- Access to models: Creating a model friendly cooperation environment
- Alignment of different understandings about collaborative work during modeling
- Empirical evidence for positive effects of modeling and model use

The aim of MoRoCo is not to discuss the advantages and disadvantages of different modeling notations. It rather puts strong emphasis in the role of models in collaborative work including their collaborative development, collaborative interaction with them as well as intertwining them with activities and artifacts of everyday work.

Accepted papers

Eight papers have been accepted for the workshop after a thorough review by an international program committee. These contributions reflect the broad scope of the workshop in contributing to a variety of aspects in the area of how models affect and are affected by collaboration.

In their paper *Cooperation on Models and Models for Cooperation* Gross and Beckmann take a theoretically based stance on collaborative model and model usage by applying Goffman's framework of social interaction to these tasks. From this perspective, they derive support needs for collaborative model usage and development.

Two papers approach the interplay of modeling and creativity and outline how creative processes can be supported methodologically and technologically. In his paper *Facilitating and Prompting of collaborative Reflection Process Models* Thomas Herrmann and Kai-Uwe Loser report on an approach to support socially distributed reflection of diagrammatic process models. They identify two fundamental concepts for reflection support: identification of model parts that can be reflected on in disjoint social groups and computer-supported prompting to substitute the role of a facilitator when reflecting in multiple groups. The paper offers examples of how these concepts could be implemented using the SeeMe modeling language and the socio-technical walkthrough. Bartelt, Vogel and Warnecke also aim at supporting the interplay between creativity and modeling within their work on *Collaborative Creativity: From Hand Drawn Sketches to Formal Domain Specific Models and Back Again*. They discuss Scribbler, a system for collaborative creation of hand-drawn models and their transformation to formal domain-specific models (e.g. based on EMF) using sketch recognition. Their toolset provides support for adaptively recognizing freehand drawings of models created by multiple users and transforming them to formal EMF-models and back again for further processing. The toolset is designed for collaborative operation and provides a set of features that support the traceability of the development history of models.

Using models to support collaboration in organizational environment almost always involves laymen modelers, i.e. people who are experts in their area of work but have no experiences in creating models or working with them. The challenge of involving these people in modeling processes is discussed in three papers presented at the workshop. In his paper *Towards Role-distributed Collaborative Business Process Elicitation*, Stefan Oppl describes an approach in letting different roles (stakeholders) in a modeling project model the processes separated from each other and how to get resulting models into a commonly agreed on model. Hoppenbrouwers, Thijssen and Vogels discuss in *Operationalizing Dialogue Games for Collaborative Modeling*, how dialogue games can be used for collaborative modeling. They present a methodology and a prototypical tool that support the process of structuring and guiding conversations for modeling. The paper focusses on the procedural guidelines necessary to implement a dialogue game for modeling and gives a good impression of how it could be facilitated. The description of the support system gives an overview of how facilitation could be aided by software. In *Beyond Collaborative Model Usage and Development – A Model Lifecycle Approach for Lay User Modeling*,

Nolte and Prilla discuss the foundations of collaborative modeling with laymen. They propose that we need concepts to engage users without modeling capabilities into self-directed, user-managed processes of using and working on models. They also presents a corresponding model lifecycle as well as suitable interaction and participation modes, using examples from their work on integrating lay-users into model usage and development.

Empirical studies on the effects of applying collaborative modeling in practical settings on the modeling process and outcome are rarely available so far. One paper that will be presented at the workshop has approached this topic. In their paper *The Added Value of Collaborative Modeling for Legal Business Rule Management* van Stokkum, Heiner, Hoppenbrouwers and Mulder describe a case where collaborative modeling is used within the area of legal modeling. Particular emphasis lies on combining business rule management with collaborative modeling in order to create a broader acceptance for new rules that are being applied. The paper thus provides an interesting and novel environment for collaborative modeling techniques.

Poppe, Recker, Johnson and Brown argue that distributed collaborative modeling requires support for visual cues used in co-located collaboration. In *Using natural user-interfaces for collaborative process modelling in virtual environments* they present their approach based on a 3D virtual world to facilitate remote collaborative process model creation and validation. However, the added complexity of having to navigate a virtual environment and using an avatar for communication makes it difficult for novice users. An improved version of a 3D modeling tool is supposed to address these issues by providing natural user interfaces for non-verbal communication, navigation and model manipulation.

Program committee

- Christian Bartelt, Clausthal University of Technology, Germany
- Eike Bernhard, Queensland University of Technology, Australia
- Sebastian Döweling, SAP Research Darmstadt, Germany
- Benjamin Fonseca, UTAD / INESC TEC, Portugal
- Stijn Hoppenbrouwers, HAN University of Applied Sciences, Netherlands
- John Krogstie, Norwegian University of Science and Technology, Norway
- Stephan Lukosch, TU Delft, Netherlands
- Jan Mendling, Vienna University of Economics and Business, Austria
- Hajo Reijers, Eindhoven University of Technology, Netherlands
- Etiënne Rouwette, Radboud University Nijmegen, Netherlands
- Barbara Weber, University of Innsbruck, Austria

References

- Baacke, Rohner, Winter, and Fitterer, (2009): 'Component-Based Distributed Modeling of Collaborative Service Processes - A Methodology for the Identification of Reference Process Building Blocks.', *HICSS* pp. 1–10, IEEE Computer Society.
- Brosch, Seidl, Wieland, Wimmer, and Langer, (2009): 'We can work it out: Collaborative Conflict Resolution in Model Versioning', *Proceedings of the 11th European Conference on Computer Supported Cooperative Work (ECSCW 2009)* pp. 207–214, Springer.
- Heer, Bostock, and Ogievetsky, (2010): 'A tour through the visualization zoo', *Commun. ACM*, 53 6, pp. 59–67.
- Herrmann, and Nolte, (2010): 'The Integration of Collaborative Process Modeling and Electronic Brainstorming in Co-Located Meetings', In G. Kolfschoten, T. Herrmann, & S. Lukosch (Eds.), *CRIWG 2010, LNCS 6257* pp. 145–160, Springer-Verlag Berlin Heidelberg.
- Klebl, Hackel, and Lukosch, (2009): 'The Application of Design Patterns for the Adaptation of a Modeling Tool in Collaborative Engineering', In L. Carriço, N. Baloian, & B. Fonseca (Eds.), *Groupware: Design, Implementation, and Use*, Lecture Notes in Computer Science Vol. 5784, pp. 262–269, Springer Berlin / Heidelberg.
- Nolte, and Prilla, (2012): 'Normal users cooperating on process models: Is it possible at all?', *CRIWG 2012, LNCS 7493* pp. 57–72, Berlin: Springer.
- Prilla, (2010): *Wissensmanagement-Unterstützung für die Erzeugung und Nutzung von Prozessmodellen als wissensvermittelnde Artefakte*, Eul Verlag.
- Prilla, Nolte, Herrmann, Kolfschoten, and Lukosch, (2013): 'Special Issue on Collaborative Usage and Development of Models', *International Journal of eCollaboration*, 9 4.
- Renger, Kolfschoten, and De Vreede, (2008): 'Challenges in collaborative modelling: a literature review and research agenda', *International Journal of Simulation and Process Modelling*, 4 3, pp. 248–263.
- Rittgen, (2009): 'Special Issue on Collaborative Business and Information Systems Design', *International Journal of eCollaboration*, 5 4.
- Rittgen, (2010): 'Collaborative Modeling: Roles, Activities and Team Organization', *International Journal of Information System Modeling and Design (IJISMD)*, 1 3, pp. 1–19.

Rittgen, (2012): 'Special Issue on Collaborative Modelling', *International Journal of Organisational Design and Engineering*, 2 1.

Yuille, and Macdonald, (2010): 'FEATURE The social life of visualization', *interactions*, 17 1, pp. 28–31.

Cooperation on Models and Models for Cooperation

Tom Gross, Christoph Beckmann

Human-Computer Interaction Group, University of Bamberg, Germany

(<firstname>.<lastname>(at)uni-bamberg.de)

Abstract. In this paper we would like to propose a Janus head perspective on cooperation and models: on the one hand cooperation on models is a very important type of activity for groups who want to create shared models that are accepted by the group members; on the other hand models for cooperation are an essential basis to develop user-centred cooperative systems.

Keywords. Cooperation on models; models for cooperation; patterns.

1 Introduction

The organisers of this workshop on ‘MoRoCo – Models and their Role in Collaboration’ at the European Conference on Computer-Supported Cooperative Work - ECSCW 2013, point out in their call for papers that ‘using visual representations for work or business processes can be considered a common practice in modern organisations. These models serve a large variety of different purposes such as documentation of current practices, or informing and planning change or software development.’ (Nolte *et al.* 2013). Indeed, models play an important role in Computer-Supported Cooperative Work (CSCW) as shared artefacts in teams that are conceived, developed, and maintained by the teams.

Besides cooperation on models, models that structure the cooperation process are an essential part of cooperation technology. Developing software that supports teams cooperating—this software is often referred to as groupware—is a challenging task and has been researched for more than two decades (Gross 2013; Marca & Bock 1992). Groupware often has a strong influence on how teams will work together. And, in fact, the effectiveness and efficiency of the teamwork as

well as the satisfaction of the individual team members strongly depend on the quality of the concepts underlying the respective cooperative technology. Schmidt (2011, p. vii) points out: ‘the development of computing technologies have from the very beginning been tightly interwoven with the development of cooperative work’. Schmidt (2011, p. vii) continues that: ‘our understanding of the coordinative practices, for which these coordination technologies are being developed, is quite deficient, leaving systems designers and software engineers to base their system designs on rudimentary technologies. The result is that these vitally important systems, though technically sound, typically are experienced as cumbersome, inefficient, rigid, crude’.

In the light of this Janus head perspective—that cooperation on models is an important part of CSCW, and that the models underlying the cooperative technology do fundamentally influence its success—this paper looks at the role of models for cooperation that can be used as basic concepts for cooperative technology that in return is used for cooperation on models. In the next section we give a brief overview of the history of models and patterns. We then introduce and suggest as a point of departure and the framework of Erving Goffman (esp. (1959)) who studied social interaction among humans and their use of their technical environment for several decades and derived a framework for social interaction. Finally, we summarise our contribution.

2 Models and Patterns

Models and patterns have a long tradition. They have early been used in architecture, most prominently by Christopher Alexander (1977). Alexander used introduced a pattern language to describe solutions that were repeatedly applied to reoccurring design challenges in the design of buildings.

Later, in Software Engineering design patterns serve a similar purpose—design patterns here have been considered as a successful approach for documenting and reusing knowledge providing a ‘way of supporting object-oriented design’ (Sommerville 2007, p. 422). Software design patterns basically have the following structure: a pattern name, a description of the problem, a description of the solution, and the consequences of the use of the pattern (Gamma *et al.* 1994).

Design patterns are also used for documenting knowledge and experience with the development of cooperative technology. Schuemmer and Lukosch (2007, p. 22) write that ‘developers building groupware applications are challenged with technical problems that are outside the focus of average software developers’. Martin and Sommerville (2004) analysed social interaction and translated their results into the format of design patterns. They (2004, p. 61) point out that ‘patterns of cooperative interaction highlight similar findings across studies related to particular socio-technical configurations, and the accompanying activities given those configurations. They start to address the question of how we

generalise from ethnographic studies to provide guidance for system designers and other users’ and ‘patterns can be of relevance and practical use to researchers and practitioners from technical or social scientific backgrounds who have an interest in social aspects of systems design’.

All these patterns provide valuable input for generating models underlying cooperative technology. And they are interesting artefacts to study when developing tools that aim at supporting teams working on them.

Yet, software design patterns primarily help structuring software, and cooperative design patterns are primarily based on the analysis of existing cooperative systems or on some ethnographical studies. In the next section we introduce Goffman’s framework of social interaction, which is based on decades of observations.

3 Goffman’s Framework of Social Interaction

Goffman’s framework of social interaction is based on decades of observations and study of related work of Goffman and provides a substantial peace of knowledge and insight into the way social interaction among humans works.

Goffman uses the metaphor of a theatre stage and points out that humans in any kind of social interaction do a performance in front of other humans who are listening and watching and interpreting the performance. Goffman writes: ‘for the purpose of this report, interaction (that is, face-to-face interaction) may be roughly defined as the reciprocal influence of individuals upon one another's actions when in one another's immediate physical presence’ and ‘a “performance” may be defined as all the activity of a given participant on a given occasion which serves to influence in any way any of the other participants’ (1959, p. 15).

His concepts that are most relevant with respect to modelling social interaction as basis for cooperative technology can be grouped into three categories: primary participants, performance, and secondary participants. Figure 1 depicts these three categories and the concepts they contain respectively.

Primary participants are humans who act according to their social status (i.e., socio-economic standing in the society). They perform a routine (i.e., a ‘pre-established pattern of action which is unfolded during a performance’ (Goffman 1959, p. 16)). According to Goffman humans have kinds of ideal interactions with each other: the optimistic ideal of full harmony (i.e., being in harmony with oneself and with others), which according to Goffman is hard to achieve; and the pragmatic ideal as a projection that should be in accordance with reality and that others can accept—at least temporarily—without showing deep and inner feelings of the self.

An interaction takes place between at least one performer and one person in the *audience*. The performer defines a situation through a projection of reality as expressions of a character bound to a certain social role in front of the audience.

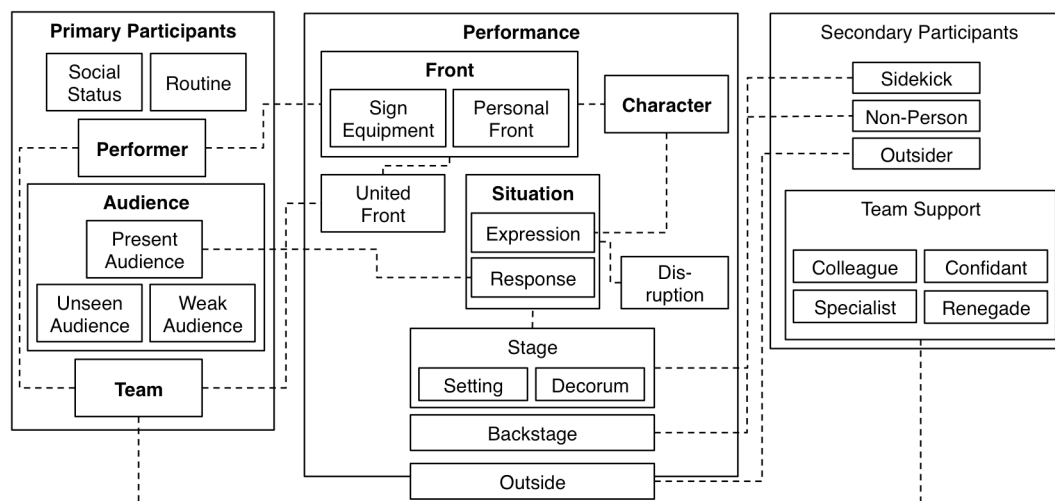


Figure 1. Concepts of social interaction from Goffman's framework of social interaction.

The performer anticipates the audience and continuously adapts the performance accordingly. Audiences can be of three different types: Present audience refers to persons who attend the performance, receive expressions, verify these in accordance to the projected situation and reality, and respond accordingly. Unseen audience are imaginary persons; the performer can use them in order to anticipate a performance. Finally, weak audience are real persons who are not present at the performance (e.g., other performers giving similar performances).

Multiple performers can act as a *performance team*. The members of a performance team need to fit together as a whole—to either present similar individual performances to amplify a projection, or to present dissimilar performances that complement to a projection.

For a performance, performers prepare a set of *fronts* shown to the audience. Fronts consist of material and immaterial parts. The material part is the sign equipment and are all properties required to give a convincing performance. The immaterial part is the personal front and refers to a performer's types of behaviour such as speech patterns. During interaction performers appear on stage through a character. A character as figure is composed of a 'front', which is adapted to the audience and performance. In a performance team, the team as whole has a united front (e.g., according to a professional status) and each member has a character with an associated front to invoke during staging.

A performance as social interaction is a finite cycle of *expressions* to define a situation and responses to feedback the validity of the expressions. Characters plays routines during performances to convey acceptable and to conceal unacceptable expressions—in a performance team multiple characters will follow this behaviour. Expressions are information that is communicated by a character, which use 'sign-vehicles' (i.e., information carriers). There are wanted

expressions that are acceptable and foster a situation as a valid projection of reality, and unwanted expressions that are unacceptable and inappropriate for a given performance in front of a particular audience. In order to manifest a performance that is coherent, a performer strives to communicate expressions consistently through their characters towards an audience. Thus a performer's character endeavours to conceal unwanted expressions. Responses are feedback from the audience, which continuously verifies the performance according to the defined situation and the overall reality as well as to the front of the character, and responds the result to the performer.

Disruptions can result from wrong or undefined projection—a consequence of a false or doubtful projection of reality based on contradictive expressions or discrediting actions. To prevent accidental disruptions a performer and an audience can agree on: the 'working consensus' as an agreement on the definition of the situation to describe a temporal value system among all participants; 'reciprocity' that means that performers guise their characters to act according to the situation (i.e., provoke neither intentionally nor factually misunderstandings) and that the audience responds to performance according to the situation (i.e., allege neither consciously nor unconsciously false behaviour); and 'interactional modus vivendi' that describes that an individual in the audience only responds to expressions that are important for the individual; the individual in the audience remains silent in things which are only important to others.

Stages provide a setting for the interaction and are embroidered with decorative properties (i.e., decorum). They support performers when fostering a situation. Both performers and the audience have access to the stage. The backstage is a region, which only performers can access to prepare and evaluate their performance. Also team members suspend backstage. The outside region denotes to neither stage nor backstage. Although it will be excluded in a performance, performers will prepare and use a dedicated front for the outside (e.g., the façade of buildings of a company).

We put other participants of Goffman's framework who are of minor important for cooperative technology in the category *secondary participants*. Participants who are involved, but are not participating in a performance are: team support (colleagues who constitute the weak audience, training specialists that build up a desirable performance and service specialists that maintain a performance, confidants that listen to a performer's sins, and renegades that preserve a idealistic moral stand that a performer or team did not kept), and sidekicks that support a single performer during performance, but in a subordinate role. Non-persons are present but neither participate nor are involved in a performance (e.g., servants). Outsiders are neither performers nor audience and have little or no knowledge of the performance. They can access the outside region; however they can invade a performance and cause a collision of performances: the outsider sees a

performance that eventually is reserved for the future when the outsider is part of the audience.

Overall, Goffman's framework provides an inspiring point of departure when conceiving of basic concepts for a model of cooperation. These concepts can be brought together in a shared model that can then—in a cooperative endeavour—be worked on in a group. The group can work on a model for any domain or business, but it can also work on a model that represents its own structure and roles of actors and ways of interaction among actors and with third parties.

4 Informing the Design of Modelling

The framework of social interaction of Goffman provides multifarious insights that have the potential to positively influence cooperation on models as well as models for cooperation.

Cooperation on models—based on the concepts above—can be characterised as follows. During the cooperation process there are typically active group members and passive group members. A group members' expressions in terms of activities can include oral or written communication, new additions to models, changes of their own parts of a model, changes of parts of the model that have been created by others, and so forth. Passive group members might watch the active person and respond (e.g., confirm that changes to their parts of the model are welcome). On the other hand the active group members might have sophisticated routines that allow them not only to concentrate on their own communication and activities, but also on the others' reactions. Active members can tightly cooperate with other active members in performance teams. The team support might include lab administrators who are responsible for maintain the distributed modelling software and hardware. Researchers have only very recently started looking at these subtleties of users' performances and others responses to them. For instance, Birnholtz et al. presented a study of collaborative writing and point out that: 'people are also concerned about how their behaviours—and they themselves—will be perceived by others' (2013, p. 961). Despite the fact that this study was on collaborative writing and not editing models, it showed interesting evidence that active users in team do care about other users responses to their performance.

Models for cooperation should use Goffman's notions as input for entities. According to Goffman several roles need to be considered by modellers of cooperative processes: performers who actively communicate and change artefacts, performance teams which consist of multiple performers, as well as audiences which can be present and visible to the active performer, unseen and weak audiences which are absent yet important. Furthermore, models for cooperation might foresee secondary participants such as team support or outsiders. In early cooperative systems and early research (cf. e.g., Rodden 1991

for an overview) the notion of a role was clear-cut to and distinct. For instance, a chair-person has specific rights and duties, and a participant has others. More recently—and in accordance with Goffman—roles have been seen as emerging and evolving over time (Finholt *et al.* 2012). Schmidt (2011, p. 31) writes: ‘the apparent stability of organizational roles and patterns of communication is a superficial hide ... Cooperative work arrangements should rather be conceived as emerging formations that change dynamically in accordance with the requirements of the situation, and cooperative work involves, inescapably, the vicissitudes of distributed decision making. These characteristics have important implications for CSCW systems design’.

As these short examples show, it is important for system designers with respect to cooperation on models and models for cooperation, to find a balance between having a structured, effective, and efficient process and providing lightweight adequate adaptability, flexibility, and spontaneity (Gross & Marquardt 2010; Schirmer & Gross 2011). This has been pointed out very early in the CSCW literature (esp. Bannon & Schmidt 1989), but neglected by some system designers.

5 Conclusions

This introduction of key concepts from Goffman’s framework of social interaction is only a starting point towards a more comprehensive discussion of key concepts—in the sense of reoccurring design patterns—of models for cooperation underlying cooperative technology. Conversely, since these key concepts and their mutual relationships can evolve into complex models it would be great to have approaches and tools to cooperatively work on them. Goffman’s framework is just one part of the overall picture; other researchers have been using other frameworks, most prominently activity theory (Kaptelinin & Nardi 1997; Nardi 1996) or distributed cognition (Hutchins 1995) (Perry 2003).

In this workshop I would like to share thoughts on how cooperation on models actually works in practice and how tools supporting this type of cooperation can be conceived, while at the same time—from a Janus head perspective—looking at the structure of this cooperation process on models and taking it as the shared artefact that the team is actually working on.

References

- Alexander, C., Ishikawa, S. and Silverstein, M. (1977). *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, Oxford, UK.
- Bannon, L.J. and Schmidt, K. (1989). CSCW: Four Characters in Search of a Context. In *Proceedings of the First European Conference on Computer-Supported Cooperative Work - ECSCW'89* (Sept. 13-15, Gatwick, UK). Elsevier, Dordrecht. pp. 358-372.

- Birnholtz, J.P., Steinhardt, S.B. and Pavese, A. (2013). Write Here, Write Now! An Experimental Study of Group Maintenance in Collaborative Writing. In *Proceedings of the Conference on Human Factors in Computing Systems - CHI 2013* (Apr. 27-May 2, Paris, France). ACM, N.Y. pp. 961-970.
- Finholt, T.A., Tellioglu, H., Inkpen, K.M. and Gross, T., eds. (2012). *Proceedings of the 2012 International ACM Conference on Supporting Group Work - Group 2012*. ACM, N.Y.
- Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA.
- Goffman, E. (1959). *The Presentation of Self in Everyday Life*. Doubleday Anchor Books, N.Y.
- Gross, T. (2013). Supporting Effortless Coordination: 25 Years of Awareness Research. *Computer Supported Cooperative Work: The Journal of Collaborative Computing* 22, 4-6.
- Gross, T. and Marquardt, N. (Sept. 2010). Creating, Editing, and Sharing Complex Ubiquitous Computing Environment Configurations with CollaborationBus. *Scalable Computing: Practice and Experience - Scientific International Journal for Parallel and Distributed Computing (SCPE)* 11, 3. pp. 289-303.
- Hutchins, E., ed. (1995). *Cognition in the Wild*. MIT Press, Cambridge, MA.
- Kaptelinin, V. and Nardi, B.A. *Activity Theory: Basic Concepts and Applications*. Presented at Tutorial #11 at the Conference on Human Factors in Computing Systems - CHI'97 (Mar. 22-27, Atlanta, GA). 1997.
- Marca, D. and Bock, G., eds. (1992). *Groupware: Software for Computer-Supported Cooperative Work*. IEEE Computer Society Press, Los Alamitos.
- Martin, D. and Sommerville, I. (Mar. 2004). Patterns of Cooperative Interaction: Linking Ethnomethodology and Design. *ACM Transactions on Computer-Human Interaction* 11, 1. pp. 59-89.
- Nardi, B.A. (1996). *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, Cambridge, MA.
- Nolte, A., Prilla, M., Rittgen, P. and Oppl, S. *Call for Papers: Workshop "MoRoCo - Models and their Role in Collaboration" at ECSCW 2013*. <http://moroco2013.files.wordpress.com/2013/05/moroco-2013-call.pdf>, 2013. (Accessed 7/5/2013).
- Perry, M. (2003). *Distributed Cognition*. In Carroll, J.M., ed. *HCI Models, Theories, and Frameworks - Towards a Multidisciplinary Science*. Morgan Kaufmann Publishers, San Francisco, CA. pp. 193-223.
- Rodden, T. (1991). A Survey of CSCW Systems. *Interacting with Computers* 3, 3. pp. 319-353.
- Schirmer, M. and Gross, T. (Oct.-Dec. 2011). Lightweight Editing of Distributed Ubiquitous Environments - The CollaborationBus Aqua Editor. *International Journal of Distributed Systems and Technologies (IJDST)* 2, 4. pp. 57-73.
- Schmidt, K. (2011). *Cooperative Work and Coordinative Practices - Contributions to the Conceptual Foundations of Computer-Supported Cooperative Work (CSCW)*. Springer-Verlag, Heidelberg.
- Schuemmer, T. and Lukosch, S. (2007). *Patterns for Computer-Mediated Interaction*. Wiley, N.Y.
- Sommerville, I. (2007). *Software Engineering* 8. Pearson Education Limited, Harlow, England.

Facilitating and Prompting of Collaborative Reflection of Process Models

Thomas Herrmann, Kai-Uwe Loser

Ruhr-University Bochum, Germany, Institute for Applied Work Science

Thomas.herrmann@rub.de; kai-uwe.loser@rub.de

Abstract. Systematical collaborative modeling usually needs a facilitator. We suggest that a large part of revising existing drafts of a process model requires facilitated reflection of what has already been achieved in the light of the experiences of the collaborating participants. This reflection can be awkward and inefficient if it takes place in a whole group of 8 to 12 stakeholders. Therefore delegating the reflection to breakout groups is reasonable but requires technically based ways of facilitation support to avoid the need to employ several facilitators. This technical support is mainly feasible for identifying reasonable segments on which a step-by-step consideration can be based, and for prompting the participants to ensure a systematic reflection.

Introduction

Collaborative modeling of business processes pursues the goal to discuss different perspectives and integrate various competences on the one hand and to make the completion of a process model more efficient. Since both goals can be conflicting, coordination is necessary as it is usually provided by a facilitator (Renger, Kolfschoten, & De Vreede, 2008; Rittgen, 2010). The facilitator provides support so that different experiences and opinions with respect to the process being modeled are taken into consideration. During the course of collaborative modeling the emerging model has to be repeatedly inspected. The inspection is a type of validation which is closely intertwined with additional elicitation of information and ongoing modeling activities. Due to the complexity of a two dimensional representation, logical dependencies, various types of relationships etc. the parts and

elements have to be deliberately reconsidered several times. A first draft of a business process model should be carefully reflected by combining the competence and experience of several stakeholders which represent various perspectives being relevant for the model under construction. This combination of several perspectives in the course of collaborative reflection leads to comparisons of diverging opinions and to negotiations of the process model, and therefore is time consuming. Consequently, it may easily happen that important issues are neglected.

These difficulties can be viewed upon from the perspective of cognitive theory: By their research on knowledge integration, Stasser and colleagues found that test persons who were required to collaboratively solve complex problems did not value relevant information which was explicitly exchanged during their discussion (Stasser & Stewart, 1992). The reasons for this behavior are not completely clarified; it is obvious that the knowledge integration of various parties requires extra effort. With respect to creativity of groups, several obstacles were identified (Diehl & Stroebe, 1987) which affect the efficiency and creativity of group work, such as production blocking, free riding, evaluation apprehension etc..

To overcome these problems, a facilitator can prompt the participants to develop new ideas and to refer to the contributions of each other and to integrate them into a shared process model. A core principle of this kind of facilitation is to visualize every participant's comments or contributions. Conklin's dialogue mapping (Conklin, 2005) can be considered as an early example of this kind of visualization.

We have developed the method of the socio-technical walkthrough with which a process model is inspected and discussed step-by step. The walkthrough method (Yourdon, 1989) is employed in many contexts to support design projects with a systematic method to reconsider the already achieved results. The systematization and the deliberate inspection of every design element and their relationships requires a facilitator who has to identify appropriate segments of a model which are inspected within one step, and who has to ensure that every segment is discussed under certain aspects. However, this kind of facilitating all cooperative interactions and visualizing their outcome may prove as very time consuming (Nolte & Prilla, 2012). In larger groups of 8-12 participants, who are usually needed to represent the relevant perspectives, the walkthrough method causes phases where most of the group members have to stay passive in a listening mode. Therefore it is reasonable to alternate the work in the whole group of stakeholders with periods of work in solitude or in breakout groups. Since some functions of a facilitator are inevitable, we propose two strategies to complement the work of a facilitator with technical functionality:

1. Support of participants to define the appropriate clusters into which the process model is segmented and where each segment becomes a subject of deliberate discussion

2. Prompting to support the reflection of selected segments by individuals or by breakout groups

The sociotechnical walkthrough

We briefly describe the basic principles of the socio-technical walkthrough (STWT) to clarify the kind of support which is needed for guiding the work of breakout groups (T. Herrmann, Kunau, Loser, & Menold, 2004; Thomas Herrmann, 2009). As Figure 1 shows the STWT is applied in a series of workshops. They take place as co-located meetings since the negotiation of diverging opinions requires a close contact between the participants. Each meeting can be used to reconsider a collaboratively modeled work or business process under one or two aspects e.g. whether the displayed activities are really necessary, how they can be supported etc. In preparation of a workshop the facilitator creates a diagram which represents the results of previous work. The facilitator develops a plan of how to inspect the complete diagram step by step. A crucial challenge is to define the segments for the single steps. If they are too small, a lot of comments of the participants will refer to aspects which belong to another segment. If the defined segments are too large it might easily happen that important details are neglected.

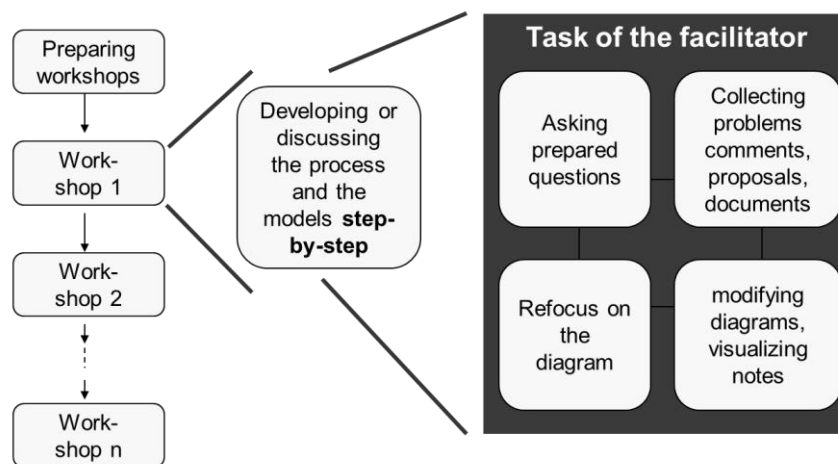


Figure 1: Process overview of STWT's

STWT-workshops are characterized by the following facilitation activities (cf. Figure 1):

- Asking prepared questions: The facilitator discloses some parts of the diagram e.g. by using hide-and-show mechanisms. Each phase of such a disclosure is one step (of about 7-15 per workshop) which is accompanied by one or two prepared questions such as: “What is the next sensible activity?”, “Which information support is needed for this activity?”.

- Collecting contributions such as answers, hints, proposals, comments, references to further documents etc. It is important that the stakeholders comment from their various viewpoints and that these contributions leave immediate traces in the process model diagram. By modifying the model, the results of the discussion are simultaneously documented.
- Focusing on the diagram: The diagram – especially the segment under discussion – serves as a focus which integrates the various experiences and perspectives of the participants into a larger picture.

In summary, the goals of the STWT are:

- Combining various perspectives, when considering the segments under several aspects (represented by questions)
- Relating every element to its context
- Reflecting the characteristics of a segment in relation to the experience of the participating experts and stakeholders.

Research on the STWT revealed that it has to be extended by means of creativity support. The linearity of the STWT is not feasible to support associative thinking and brainstorming (Thomas Herrmann, Nolte, & Prilla, 2013).

In the following we want to discuss and propose how the STWT-oriented collaboration can become more efficient, if the walkthroughs are delegated to breakout groups. For instance, with three breakout groups a model could be discussed and modified under three different aspects. In such a constellation it is not reasonable to engage three facilitators but to technically support the groups themselves to run a systematic walkthrough.

Support of segmentation

A first measure is to support the groups to define the segments – under which they intend to walk through the model – by themselves. This can happen by asking the members of the whole group to identify for every element of a process model which other two or three elements of the model are most closely related to them – from a semantic point of view.

To demonstrate this we ran a first small explorative study. We asked eight people to identify relations between the sub-elements shown in Figure 2: “The elements of this diagram are labeled with differently colored points. Please add points of the same color to two other elements which you consider as closely related to the element with the same color”. However we did not show them the nested structure of the model to avoid a pre-orientation on certain clusters. The results of eight people’s proposals for defining relationships between the elements were manually entered into the model by establishing directed relations and annotating their cardinality depending on how many participants have indicated the relationship. At

the first glance, nearly every element was connected with more than 5 other elements.

Processing and coordination of service requests

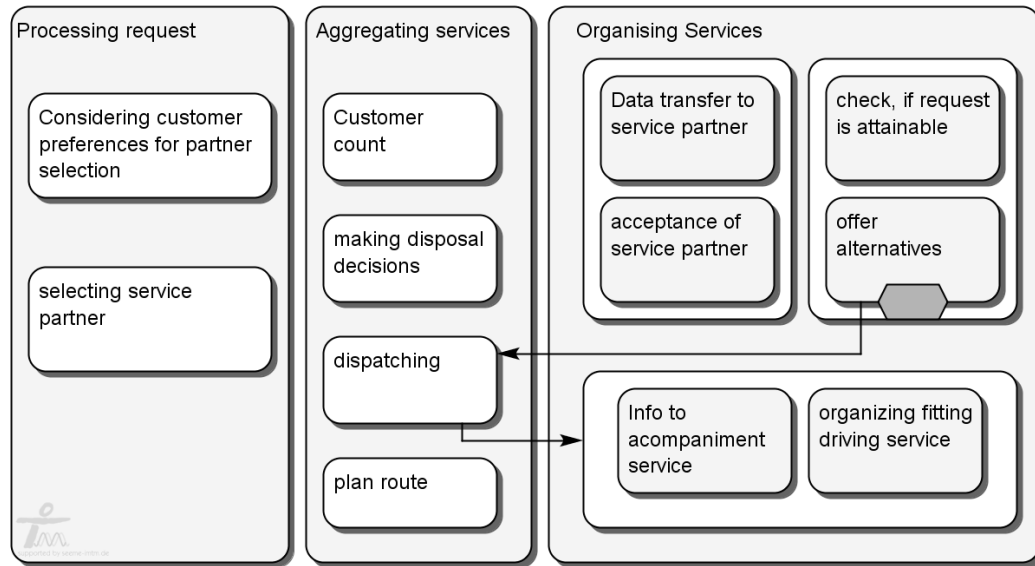


Figure 2: Part of a diagram for which reasonable segments had to be identified

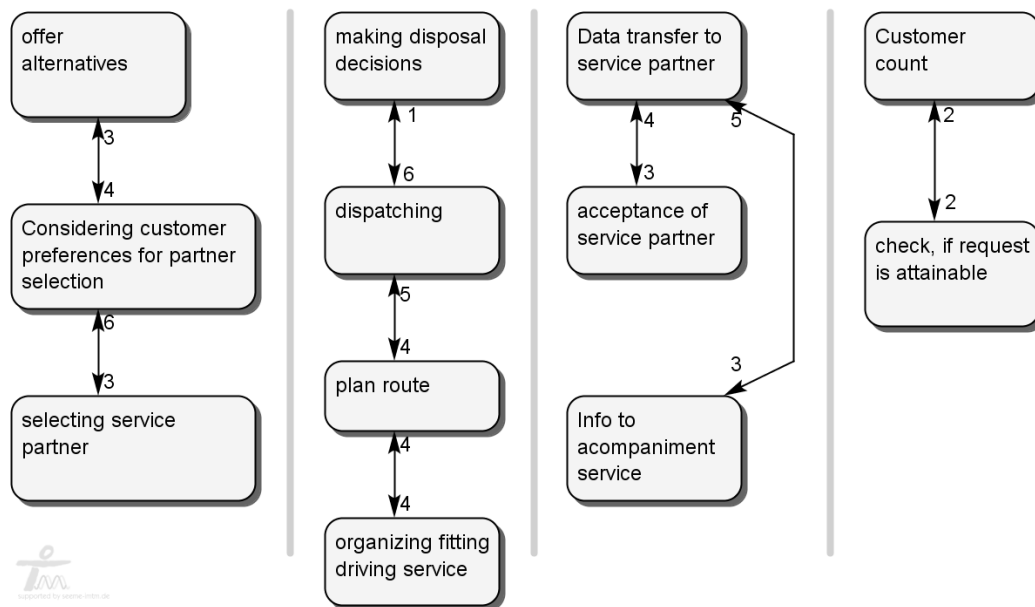


Figure 3: Results of collaborative identification of segments of a part of a process diagram

To make a structure of segmentation visible we carried out the following operations:

1. All relations are weighted by the number of their occurrence (see Figure 3). For this purpose, the counts of the two directions of a relation are added.
2. All relations of a weight of N are deleted, starting with $N=1$,
3. The deletion of a relation is not conducted if this deletion causes that an element remains without any relation to the others.
4. N is increased until no deletion can be carried out.

Surprisingly, the resulting clusters do not match the clusters being provided by the nesting structure in Figure 2. The super-elements (such as “processing request”) are usually proposed by a modeler or the facilitator. Usually the nesting structure is employed to define the segments of the walkthrough. The experimental study revealed that this strategy might not be always appropriate. The tested method of building segments also revealed that the suggested semantic relationships of a drafted model might need to be revised. Further research will have to deal with an extended functionality which helps to handle models with a larger, realistic number of elements and supports the automatic identification of appropriate clusters to define the steps of a walkthrough.

Support of prompting

One important task of the facilitator is to provide prompts which stimulate the participants to reflect the status of a process model and to make contributions. Appropriate prompting is discussed as a method to increase the creativity level of facilitated brainstorming (Santanen, Briggs, & de Vreede, 2004). From a cognitive view, prompting can help to overcome the linearity of thinking and to combine the relevant aspects of a process in unusual ways (T. Herrmann, 2009). Furthermore, prompting has been widely researched in the context of learning and teaching, especially for computer supported collaborative learning (CSCL). Prompting (Thillmann, Künsting, Wirth, & Leutner, 2009) can be seen as a part of scaffolding which mostly consists of a guidance through a procedure which combines several mandatory and optional activities. The STWT is an instance of such a procedure. The prompts remind people to not forget steps which might be helpful in certain situations. CSCL-research pursues the concept to provide those prompts by technical functions during human-computer interaction which help the collaborating participants to conduct important steps in the process of learning.

We have applied the research on prompting in the context of supporting reflection at the workplace (Prilla, Degeling, & Herrmann, 2012); the intention is to guide people to articulate their experience with certain work situations by either describing the situation or noting down the result of their reflection. Subsequently, these articulations can be shared with other people who made similar experiences. The

interaction with others may help to find solutions and to support each other to bring these solutions into reality.

With respect to the socio-technical walkthrough, the following activities could be prompted:

- The leading question can be repeated for each segment;
- Participants can be asked for their opinion;
- “What-if-“ or “what-else-“ questions can be used to stimulate creativity;
- Participants are reminded to leave tracks of their discussion in the model;
- After each modification the collaborators can be asked to declare whether they agree with it;
- The participants can be asked to see the segment under discussion and its modification in the context of the whole process model;
- The collaborators can be asked whether they agree to proceed with the next segment.

By delegating this prompting to the technical functionality, the participants do not have to care by themselves about the systematization and coordination of the walkthrough but can focus on the content of the collaboratively modeled process in relation to their expertise.

Summary: Reflection support for collaborative modeling

All in all the described concepts for support of collaborative modeling can be related to research which intends the support of reflection at work. Selecting an appropriate unit, to which reflection refers, focusing on it without neglecting the larger context and continuous prompting which avoids the neglecting of important aspects of the participants’ perspectives and of documenting the results can be considered as relevant principles which should be technically supported. This helps to conduct systematical reconsideration and negotiation of drafts during collaborative modeling in breakout groups without employing a facilitator for each group. Besides the use for STWTs, it might also be possible to use the support for other types of collaborative work on artefacts. Further research has to prototype solutions for this kind of support and to run experiments to refine these solutions for interactive identification of segments and for appropriate prompting. The main technical challenge with respect to prompting is to make it as unobtrusive as possible and to adapt it to the users’ needs for scaffolding. Other aspects for research are to consider the limitations of knowledge integration if work on models is delegated to break out groups which only include a reduced scope of perspectives. Therefore, appropriate means of facilitation methods have to be identified to bring the perspectives of several breakout groups together.

References

- Conklin, J. (2005): Dialogue mapping. Wiley, Chichester.
- Diehl, M., & Stroebe, W. (1987): 'Productivity loss in brainstorming groups: toward the solution of a riddle'. *Journal of personality and social psychology*, 53(3), pp. 497–509.
- Herrmann, T. (2009): 'Design Heuristics for Computer Supported Collaborative Creativity'. *System Sciences*, 2009. HICSS'09. 42nd Hawaii International Conference on (S. 1–10).
- Herrmann, T., Kunau, G., Loser, K. U., & Menold, N. (2004): 'Socio-technical walkthrough: designing technology along work processes'. *Proceedings of the eighth conference on Participatory design: Artful integration: interweaving media, materials and practices-Volume 1*, pp. 132–141.
- Herrmann, Thomas. (2009): 'Systems Design with the Socio-Technical Walkthrough'. In B. Whitworth & A. de Moor (Hrsg.), *Handbook of Research on Socio-Technical Design and Social Networking Systems*. Information Science Reference.
- Herrmann, Thomas, Nolte, A., & Prilla, M. (2013): 'Awareness support for combining individual and collaborative process design in co-located meetings'. *Computer Supported Cooperative Work (CSCW)*, 22(2), pp. 241–270. doi:10.1007/s10606-012-9179-x
- Nolte, A., & Prilla, M. (2012): 'Normal users cooperating on process models: Is it possible at all?'. *CRIWG 2012, LNCS 7493* (S. 57–72). Berlin: Springer.
- Prilla, M., Degeling, M., & Herrmann, T. (2012): 'Collaborative Reflection at Work: Supporting Informal Learning at a Healthcare Workplace'. *Proceedings of the ACM International Conference on Supporting Group (GROUP 2012)*.
- Renger, M., Kolfschoten, G. L., & De Vreede, G. J. (2008): 'Challenges in collaborative modeling: a literature review and research agenda'. *International Journal of Simulation and Process Modelling*, 4(3), pp. 248–263.
- Rittgen, P. (2010): 'Collaborative Modeling: Roles, Activities and Team Organization'. *International Journal of Information System Modeling and Design (IJISMD)*, 1(3), pp. 1–19.
- Santanen, E. L., Briggs, R. O., & de Vreede, G.-J. (2004): 'Causal relationships in creative problem solving: comparing facilitation interventions for ideation'. *Journal of Management Information Systems*, 20(4), pp. 167–198.
- Stasser, G., & Stewart, D. (1992): 'Discovery of hidden profiles by decision-making groups: solving a problem versus making a judgement'. *Journal of personality and social psychology*, 63(3), pp. 426–434.
- Thillmann, H., Küsting, J., Wirth, J., & Leutner, D. (2009): 'Is it Merely a Question of “What” to Prompt or Also “When” to Prompt?'. *Zeitschrift für Pädagogische Psychologie*, 23(2), pp. 105–115.
- Yourdon, E. (1989): *Structured walkthroughs*. Prentice-Hall Englewood Cliffs, NJ.

Collaborative Creativity: From Hand Drawn Sketches to Formal Domain Specific Models and Back Again

Christian Bartelt, Martin Vogel, Tim Warnecke
Software Systems Engineering - Department for Computer Science
University of Clausthal, Germany
{christian.bartelt, m.vogel, tim.warnecke}@tu-clausthal.de

Abstract. Most of the time developers make extensive use of software tools in a software development process to support them in their day-to-day work. One of the first and most important phases is the design phase. Here tools are missing which support the creative and collaborative workflow (parallel/distributed). At the moment software designers use classic whiteboards in team meetings to express their ideas. Subsequently a coworker uses a mobile phone or a camera to take photos of the work and remodel the picture with a modeling tool. That process is very inconvenient, error-prone and hinders a creative modeling cycle. For overcoming this ineffective process this paper shows a new approach to use digital whiteboards to transform free hand sketches in formal models and back again while modeling in a distributed team. The approach is completely independent from a pre-defined modeling language. It provides an interactive training mode to learn new graphical syntax elements and map these elements to formal meta-model entities. Based on the approach a collaborative sketch and modeling infrastructure was implemented.

Video: <https://www.youtube.com/watch?v=0i3M9djPrRM>
[Mirror: <http://sse-world.de/index.php?clD=3611>]

Motivation and State-of-the-Art

Nowadays, software development is a creative and distributed team process. The early and creative design phase is very important for successful software projects.

Normally software designers do not use modeling tools like MagicDraw UML (MagicDraw 2013) in this early phase because of their inconvenient operation. Modeling tools are made for precise model design, but not for creative sketching. That is the reason why software designers using whiteboards in team meetings to visualize and communicate their ideas within the group (Cherubini et al. 2007). Nevertheless, after a successful team meeting, one of the designers has to transform the drawn sketches in formal models using the previously rejected modeling tools. Those transformations are error-prone because for e.g. the designer tries to optimize the diagram or forgets some elements. Thus it is possible that the new formal diagram cannot be recognized by the other team members in a later meeting because of its changed appearance.

Another widely known problem in this domain describes that everybody has to be present to accomplish such a creative meeting. If a team is spread all over the country or worse over the world it needs a lot of effort, money, and time to get them all to one place. One possible solution is to use Screen Sharing Software like TeamViewer (TeamViewer 2013) to share some kind of drawing software and additionally utilize a telephone conference system. An issue here is that two or more software applications are used together and none of them is particular conceptualized for software designers.

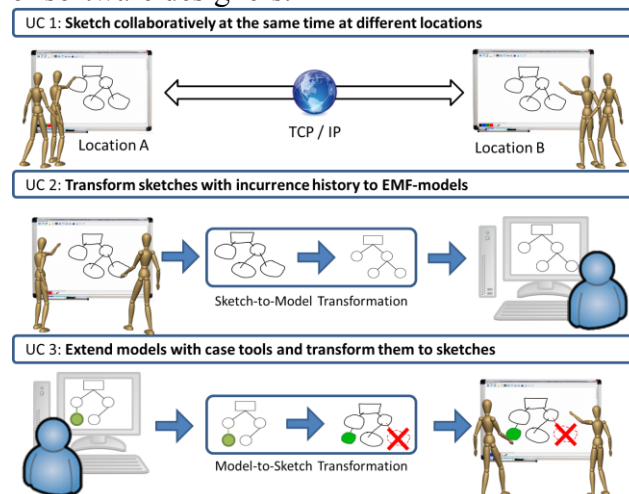


Figure 1: Use cases

In Figure 1 three typical modeling use cases in the creative engineering phase are depicted. The first use case (UC 1) describes two distributed teams are working parallel on the same sketch model. UC 2 shows how a sketched model, which was drawn by a team of designers and transformed to a formal domain specific model. This transformed model is based on Ecore, which is now usable with a corresponding EMF- or GMF-Editor (whereas the whole history of origins of the sketch is preserved). This allows a single user to extend and change the model based on the results and feedback of the previous team meeting. UC 3 illustrates how such a modified model is transformed back to a sketched model.

Because Scribbler knows how the sketched model looked before all changes made in the GMF-Editor can be visualized, e.g. an element was deleted or added.

The last both use cases make it possible to establish a creative life cycle which starts with a onetime configuration. This configuration consists of a knowledge base, which contains of previously learned sketches for the chosen DSL, and a mapping between those sketches and elements of the DSL meta model (Ecore). It should be noted, that this configuration can also be done after drawing sketches. After the team drew some sketches the sketch model is transformed, with help of the configuration, to a formal model (UC 2) which is suitable for a GMF-editor. In this editor a single user modifies the model based on the feedback of the meeting before. After he finished his modifications, the model is transformed back to a sketch model and given back to the team (UC 3). All modifications done in the GMF-Editor are visualized and the remaining, but unchanged, sketches look like in the first meeting. Thus the team is able to better recognize what happened in the last meeting. To improve the process of recognizing a sketch, which is made some days or weeks before, the designer can also use the implemented history viewer which recorded every stroke made on the canvas.

With similarities to these use cases in the last few years several research initiatives are started with the topic intuitive modeling respectively model sketching. In (Sangiorgi & Barbosa 2010) a recognition mechanism for sketched model elements was presented. This mechanism uses a similarity calculation between drawing traces based on the Levenshtein distance (Levenshtein 1966) and is also a foundation for the research here explained. Some innovative research results about sketch recognition in the area of requirements modeling was described in (Wüest, Seyff, and Glinz 2013). Some further recognition techniques based on vector comparison between sketches and GEF/GMF model elements were published in (Scharf 2013).

Scribbler – The Collaborative Sketching/Modeling Infrastructure for Domain Specific Models

The developed sketching/modeling platform Scribbler picks up the upon explained use cases. Therefore, it must overcome several challenges like sketch recognition, formal model synthesis from recognized sketches, sketch synthesis from formal model, and an efficient mechanism to allow distributed parallel sketching. Furthermore, the approach should be easy/user friendly adaptable for sketching any kind of domain specific syntax.

Sketch Recognition and Knowledgebase

A sketch is per definition a freehand drawing, consisting of some individual elements, which is not yet finished and tries to transport some kind of idea (Davies 1990). Following from this a sketch and its elements has meanings for the people who work with it, but for computers sketch elements are only a set of x- and y-coordinates and maybe colours. These coordinates must be interpreted in a way such the computer knows what the drawing person had in mind. Aggravated by the fact that every human being sketches figures a little bit different, four main problems must be taken into account when recognizing a figure. First the drawing order of a figure alters between different users, second the size of a figure changes with attempt, third a figure can be inclined to the right or left side and last users often don't draw solid lines.

To solve these problems, Scribbler uses an extended and modified version of an algorithm described by (Coyette et al. 2007). In a first step of the procedure a sequence of numbers based on the intersections with a predefined grid is produced. In the example in Figure 2 the sequence of the circle is *1 1 2 3 4 5 8 9 12 13 14 15* because two intersections are detected in field 1, one additional intersection in field 2, one additional intersection in field 3 and so on.

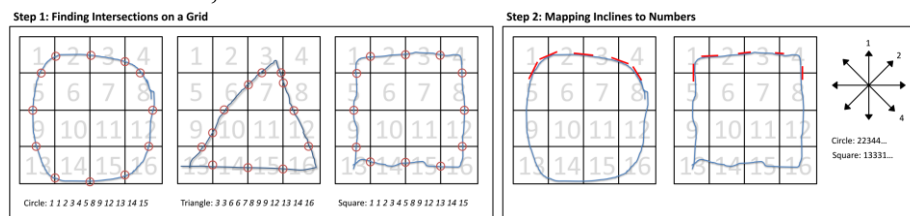


Figure 2: Sketch Recognition

This is pretty straight forward, but not every drawn object has a unique numerical sequence. A circle could have the same sequence as a square because they have the same intersections with the used grid. In a second step the incline for every intersection of the drawn line is measured and mapped to a number corresponding to the scheme shown on the right side of Figure 2. This generates a second numerical sequence for both sketches, which is now completely different.

After constructing such a pair of sequences a knowledgebase of previously drawn figures is needed to compare them with new drawn sketches and finding a match using Levenshtein distance (Levenshtein 1966). Building such a knowledgebase needs some kind of learning environment, which is done in Scribbler with an own dialog.

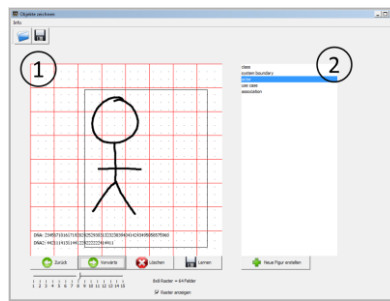


Figure 3: Learning environment

This dialog is shown in Figure 3 and consists of a training canvas (1) and a list of elements (2) which were already learned. The user is also able to add new versions of a sketch by redrawing it over and over (leads to better recognition results) again and to create new sketches by adding them to the list. Every drawn sketch will be automatically added to the knowledgebase. The knowledgebase is stored in a file, which can be exchanged with other users. For collaborative work recognizing of sketches is important, because the team gets a visual feedback of what happened. If the feedback is not that result which they actually discussed, the team can react immediately and change the type of the sketch to the desired one.

From Sketches to Formal Models and Back Again

Transforming a hand drawn sketch to a formal Ecore (EMF 2013) based model describes a process of mapping sketches to elements of the given meta model. At first glance this sounds easy but a lot of information, like for example position, size and the history of origins, is lost if hand drawn model elements are just mapped to their corresponding EMF counterparts. Transforming the EMF model back to a hand drawn sketch is not possible any more without theses information.

Due to this problem a new file type was needed. In this file three types of information are stored at the same time for every element. The first is the meaning of an element defined by the corresponding meta model. The second is the graphical representation given by the Graphical Modeling Framework (GMF 2013) such as, for example a UML class is a rectangle with an additional horizontal stroke. The last type of information stored the coordinates of the hand drawn strokes and the history of origins logged by Scribbler. This new file type sets Scribbler in the position to transform hand drawn sketches to ECore based models and back again without loss of information.

Collaboration: Drawing together and saving the history of origins

Another problem with hand drawn sketches is that the incurrence of its elements is not traceable anymore. This is especially problematic if a sketch was drawn some days or weeks ago and someone tries to remember what happened in the

meeting (Cherubini et al. 2007). Scribbler solves the problem by saving all drawing actions which happened on the canvas in a file for later use. The core of it fires for every user action an event, e.g. drawing or moving. Events are stored in a queue for the plugins. This queue is stored in a file with the whole sketch model. Thus, the history of origins still remains. For the transformation from sketches in formal models, the history is also stored in the new model file. After this step it is possible to review the whole drawing process of the sketch in a user defined speed and to stop and restart it whenever the user wants to.



Figure 4: History viewer

Figure 4 shows the user interface of the history viewer. It looks like an audio/video tool with a play button and a slider for the timeline. Thus, the history of origins of the model can play back. This feature is important for collaborative modeling, because if a new member joins the team, he can comprehend how the model is originated in the team. He can jump to every position in the origination process. Further the history viewer might be used in future for contextual modeling. This means that the team navigates to the position, which they want to modify and change the information in the sketch. After the modification they navigate to the end of the timeline and continue the work at the model. The modification saved in the history of origins.

Another component of Scribbler is the collaboration platform. Since Scribbler is an intuitive modelling tool, which is inspired by a standard whiteboard, it is necessary to construct a collaborative lock-free environment in which everyone immediately sees if a user starts a new sketch, how he draws it and the name of the user. Implementing such a collaborative environment is a challenge because every client doesn't use necessarily the same hardware and software. This fact leads to two new problems. Different devices have different screen resolutions and bandwidths.

Solving problem number one, Scribbler scales up every drawn sketch to a fictional resolution and scales them down to the actual resolution of the corresponding device. This procedure ensures that every screen size is supported no matter how big or small it is. Problem number two is solved by transferring only mouse movements/events and the coordinates, so no screen sharing is necessary. Furthermore, the server saves every draw session. This feature allows sending all transmitted data to a new connecting client and pushing him to the current stage of work. Additionally the server has the ability to save the cached data of a session in a local file. Thus it is possible to load and continue such an older session or view it in the history viewer. Thereby every member of a team can prepare for the next meeting or evaluate the session afterwards.

Tool Implementation

Scribbler is just a simple paint program which supports only basic operations like draw, move, scale, delete, save and load. Every drawn sketch consists of a series of raw dates, like, for example, coordinates and mouse movements. Scribbler gains its sketch recognition and collaboration skills through plugins.

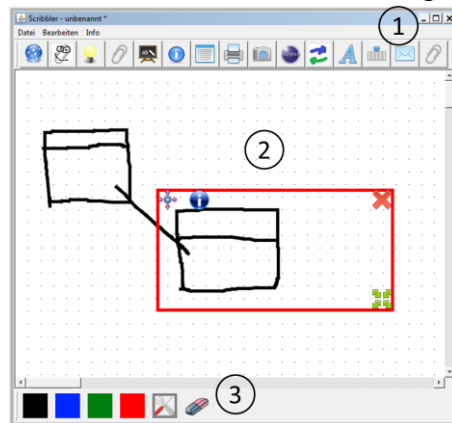


Figure 5: Tool Scribbler

A screenshot of the tool is shown in Figure 5. At the top of it (1) all current loaded plugins are represented with an icon. In the center (2) is the canvas and last but not least the toolbar is located at the bottom (3), which consists of four colors, an edit button and a rubber.

Evaluation

During the project duration three industry partners from different domains used the Scribbler for their daily work – with customers and for architectural and structure meetings - for about four weeks. Every team had an experimental setup composed of a digital whiteboard and some tablet pcs and a catalogue of questions to evaluate the Scribbler. The results of the evaluation as described below. The three teams used the Scribbler only for collaborative work – especially the history viewer and the server sessions - to prepare the next meeting. The teams assessed this functionality as valuable and it is very helpful for their daily work. Furthermore, they used the learning environment to insert their own elements for their own domain languages. Scribbler was able to learn all of these elements and the recognition rate was very good. Concerning the usability and the training period every participant rated the Scribbler as good.

Conclusion

Ensuing from the requirements regarding an intuitive modeling infrastructure that does not hinder the creative engineering process the sketching/modeling platform Scribbler was developed. Scribbler allows a distributed, parallel (collaborative) sketching of engineering models on digital whiteboards, the transformation of sketches in (semi-)formal domain specific models and back again, an easy and interactive learning of new domain specific syntax elements, and a recording/playback of the modeling/sketching history. The fundamental concepts of all these features are explained in this paper. Furthermore the implemented software infrastructure is presented. For future work the recording of further context information during the sketching modeling process (e.g. voices of modelers within the history of model evolution etc.) is planned.

This research work was supported by “German Federal Ministry of Education and Research” (BMBF) within the Project “KoMo – From Sketch to Model: Cooperative Modeling with Domain Specific Languages” (2011-2013).

References

- Cherubini, Mauro et al. 2007. «Let's go to the whiteboard: how and why software developers use drawings» in . *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 557–566. CHI '07. New York, NY, USA: ACM.
doi:10.1145/1240624.1240714. <http://doi.acm.org/10.1145/1240624.1240714>.
- Coyette, Adrien et al. 2007. «Trainable sketch recognizer for graphical user interface design» in . *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction*, 124–135. INTERACT'07. Berlin, Heidelberg: Springer-Verlag.
<http://dl.acm.org/citation.cfm?id=1776994.1777013>.
- Davies, Diana. 1990. *Harrap's Illustrated Dictionary of Art and Artists*. Chambers.
- EMF. 2013. «EMF». www.eclipse.org/modeling/emf/.
- GMF. 2013. «GMF». www.eclipse.org/modeling/gmf/.
- Levenshtein, Vladimir I. 1966. «Binary Codes Capable of Correcting Deletions, Insertions and Reversals» in . *Soviet Physics Doklady*, 10:707 – 710.
- MagicDraw. 2013. «NoMagic - MagicDraw». www.nomagic.com/products/magicdraw.html.
- Sangiorgi, Ugo Braga & Barbosa, Simone D. J. 2010. «SKETCH: Modeling Using Freehand Drawing in Eclipse Graphical Editors» in . *Proc. FlexiTools Workshop*.
doi:<http://www.ics.uci.edu/~tproenca/icse2010/flexitools/papers/10.pdf>.
- Scharf, Andreas. 2013. «Scribble - A Framework for Integrating Intelligent Input Methods into Graphical Diagram Editors» in . *Software Engineering 2013 Workshopband (inkl. Doktorandensymposium)*, 591–596. <http://www.se2013.rwth-aachen.de/downloads/proceedings/SE2013WS.pdf>.
- TeamViewer. 2013. «TeamViewer». www.teamviewer.com.
- Wüest, Dustin; Seyff, Norbert & Glinz, Martin. 2013. «FlexiSketch: A Mobile Sketching Tool for Software Modeling» in Uhler, David et al. (arg.). *Mobile Computing, Applications, and Services*, 225–244. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 110. Springer Berlin Heidelberg.
http://link.springer.com/chapter/10.1007/978-3-642-36632-1_13.

Towards Role-distributed Collaborative Business Process Elicitation

Stefan Oppl

Department of Business Information Systems – Communications Engineering, Kepler University of Linz, Austria
stefan.oppl@jku.at

Abstract. Elicitation of business process knowledge can be facilitated by conceptual models of collaborative work. Models of collaborative business processes with actors participating in different roles are complex constructs with flows of individual activities that are coupled via acts of communication. The process of elicitation in such cases can benefit from separating the modeling process for each role and let actors focus on their own contribution to work and their communication with other roles. This paper identifies concepts for model elicitation and modeling support that enable a modeling process distributed across roles and identify collaboration issues while maintaining one consistent overall model representation. A modeling methodology implementing these concepts is presented and first results of exploratory tests are discussed.

Introduction

In the last years, business process models have become a recognized means for representation of knowledge about collaborative work in organizations (Gasson, 2005). They can be used for communication of information about work and facilitate elicitation and alignment of business process knowledge (Rittgen, 2007). The creation of sound and fully specified business process models, in addition, allows to go beyond communication support and enables further processing like validation, optimization and execution of the model (Giaglis, 2001).

Business process models are a representation of organizational work with activities distributed over different actors. Elicitation of information about work thus

has to involve all relevant stakeholders to form a comprehensive model of the work process Antunes et al. (2013). and focus on collecting information from the actual workers to avoid intermediate expert modelers, who may lack tacit knowledge about the actual implementation of work (ibid.). Involving the stakeholders during modeling confronts them with different viewpoints and conceptions of how the collaboration should be implemented (Stuit et al., 2011), which need to be aligned in the process of creating the model.

The goal of the research the present work contributes to is to enable stakeholders to directly create models of business processes without intermediaries and facilitate the uncovering and eventual resolution of different conceptions of the collaborative work process. At the same time, the resulting process models should allow for formal validation and further processing, e.g. in workflow support systems. Similar objectives have been targeted in earlier research (e.g. Herrmann et al. (2007) or Rittgen (2010)). The approach presented here follows a different approach by letting stakeholders focus on solely their individual role in a process (i.e. their activities and communication with others) in contrast to existing works, where an overall view on the whole business process is maintained for all stakeholders. Focusing on the individual contribution to a work process leads to more detailed and refined models that better reflect the actual perception of their work (Dann, 1992) and also enables to explicitly identify different conceptions on the need for communication during work. The objective of the present paper is to explore approaches for model elicitation that enable capturing process knowledge separately for each involved role and support the identification of conflicts in the perceived work process and facilitate the resolution of this issues.

In the next section, the process elicitation approach is described. The following section outlines the requirements on methodological support during elicitation. Focussing on methodology, the subsequent section introduces the concept of role-distributed modeling and describes three different ways of creating role-distributed models. The final section briefly reports on the first in conducting role-distributed modeling and outlines the next research steps.

Elicitation Approach

Depicting collaborative work in business process models requires a clear understanding of the concepts relevant for modeling. Following the approach of role-centric modeling, i.e. describing who is doing what and communicating with whom in the course of collaborative work, the relevant concepts used in the area of business process modeling are “actor”, “role”, “activity” and “communication” (Soderstrom et al., 2002).

Actors are individuals who are actively participating in a work process. Activities of different actors are carried out in parallel without immediate interaction with others and are coupled via explicit acts of communication (i.e. transferring work results from one actor to another). Decisions on which activities are carried out from a number of options are made by the actor based upon the outcome of a

prior activity or the content of incoming communication.

Business processes are not only valid for one specific set of actors but are specified in a more abstract way for a set of interacting roles. A role is an area of responsibility in a business process. Consequently, several actors are able to take a certain role in a business process. Communication acts are carried out among roles and interlink the activities carried out by actors acting in a certain role.

When designing support for eliciting knowledge about work processes, the different kinds of activities described above have to be considered as fundamental model elements. We distinguish the following types of activities: (a) individual activities carried out by an role (including decisions); and (b) communication acts to link individual activities of different roles: (b1) outgoing communication acts, i.e. actively sending work results or (b2) incoming communication acts, i.e. receiving work results.

A modeling language used to support role-distributed business process modeling has to provide constructs that allow for structuring the model along role boundaries in order to allow for visualizing a model distributed along the involved roles and keep the model parts interrelated (Adamides and Karacapilidis, 2006). Modeling languages using the “role”-concept or equally interpretable constructs as the primary factor of structuring meet this requirement (Giaglis, 2001). UML Activity Diagrams (de Cesare and Serrano, 2006), BPMN (White, 2004), or S-BPM Fleischmann et al. (2012) are examples of business process modeling languages that enable this structuring approach and at the same time have existing tool support for validation or execution.

Separating a process along the involved roles has implications for modeling support. Modelers need support for interlinking and aligning different contributions to a business process and ultimately deriving a commonly agreed upon model of the business process. Each role’s contribution to work is created as a separate part of the model. As noted above, one role can be taken by several actors in an organization. Different actors introduce different viewpoints about how one role’s contribution can be implemented (Herrmann et al., 2002). These different viewpoints require alignment to derive one single, commonly agreed upon view on a business process. Consequently, an elicitation instrument has to support *collaborative modeling of role behavior*. All participating actors in this case share the same part of the model.

The role-based process parts are interconnected by communication acts, which are represented by flows of discrete messages. The following activities can occur in modeling communication (using the concept “message” to represent transmitted results of work): (a) send a message to another role; (b) get notified that a message has been sent to one’s own role; (c) request a message from another role to be able to proceed with one’s own part of the process; and (d) get notified that another role requests a message to be able to proceed with its part of the process.

The first two communication acts (a and b) are sufficient to describe all communication situations if the business process is modeled in fully sequential manner. This, however, requires actors to wait for another role to send a message before they can proceed with modeling their own process part. Communication acts c and

d are introduced to avoid these delays in modeling and to explicitly allow to express expectations on modeling that might require further discussion. *Elicitation support has to allow the specification of these different types of messages as well as the resolution of inconsistent communication acts across roles.*

Support for Role-distributed Elicitation

A role-distributed modeling support concept is presented here to explore methodological options to meet the requirements described above. A simple modeling language is used for this purpose, following the minimal requirements on a modeling language supporting role-distributed modeling as specified above. Three different types of modeling elements are used:

Activity modeling elements are used for representing activities carried out by a role as well as acts for sending and receiving messages. The semantics of the element (i.e. do something, send, receive) is determined during modeling time. Message elements are used to either send a message (outgoing message element) to another role or request a message from another role (incoming message element). Their respective incoming or outgoing message counterparts are added to the communication partner's modeling surface to link. Incoming messages or message requests, however, do not necessarily need to be processed by the communication partner immediately. They are pooled in tray areas that visualize all unprocessed messages (cf. Figure 1).

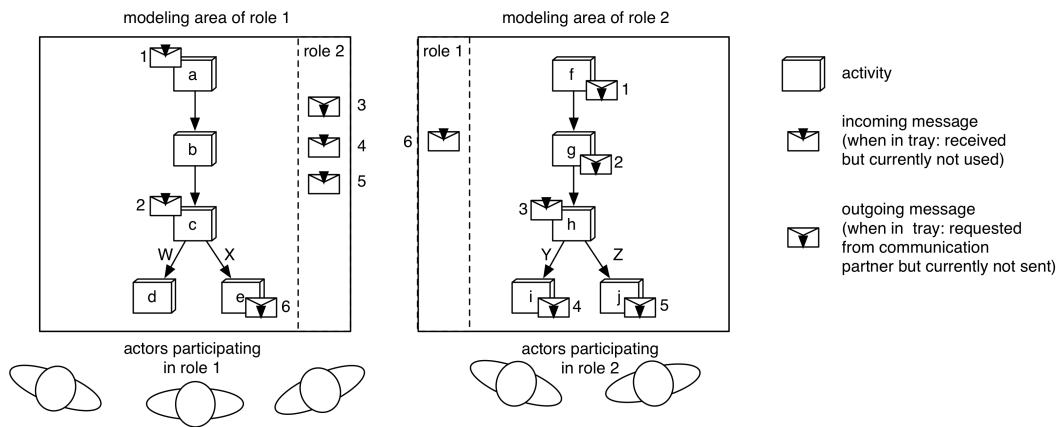


Figure 1. Example setting of role-distributed models in an intermediate stage during modeling.

The use of the three modeling elements are visualized in Figure 1, which shows an elicitation process in an intermediate stage for illustration purposes. The depicted scenario consists of two interacting roles. The behavior of role 1 is modeled by three actors, two actors provide input for role 2. The modeling surfaces include trays for coupling to the respective other role on one of their borders.

Activities (labelled with lower-case letters in Figure 1) are placed on the surface and are associated following their sequential order. Optional paths are represented

by decision parameters placed next to the according association link (labelled with upper-case letters in Figure 1).

The two model parts are interlinked using message elements (labelled with numbers). Following the coupling concept, message elements always exist in pairs of two. The semantics of a message element changes depending on whether attached to an activity element or kept in the tray area: (a) an incoming message attached to an activity (e.g. activities a, c, or h in Figure 1) represents the act of processing a received message; (b) an outgoing attached to an activity (e.g. activities e, i, or j in Figure 1) represents the act of sending a message to a communication partner; (c) an incoming message placed in an tray area represents a message that is offered by a communication partner, but has not yet been processed; and (d) an outgoing message placed in an tray area represents a message that is expected by a communication partner, but has not yet been created and sent.

The messages kept in the tray areas make mutual expectations and potential communication flaws explicitly visible. Requested messages or unused incoming messages that remain in one of the trays always point at a mismatch between the expectations and the current behavior of the communication partners. During elicitation, this visualization of communication problems triggers negotiation and alignment activities that allow for the specification of a sound overall model.

Three different procedural approaches for distributed model elicitation can be identified following the concept of behavior and communication specification described above. They differ in the point in time when message specification happens. In *ex-ante communication negotiation*, all messages are specified collaboratively by the involved actors before the roles' behaviors are described. The messages are initially placed in the tray areas for each role and are then used during behavior modeling. In *ex-post communication negotiation*, each role's behavior including all outgoing and required incoming messages are modeled separately. In a consolidation step, the communication among the roles is then aligned by mutually matching requested and sent messages. In *ongoing communication negotiation*, messages are put into the trays of communication partners immediately as they are specified during behavior modeling. Inconsistencies or different understandings are discussed immediately.

All three approaches stress different aspects of the modeling process and appear to be suitable for different modeling purposes. *Ex-ante communication negotiation* creates a common overall picture of the work process to start with and leaves identification of communication problems to the subsequent distributed modeling phase. Uncovered communication problems then require an additional round of alignment. *Ex-post communication negotiation* by contrast forces modelers to initially only focus on their own contribution to the work process. The identification of inconsistent communication acts is most likely here. The alignment of communication acts could lead to the need for a subsequent revision of roles' behavior models, if fundamental inconsistencies, e.g. conflicting communication sequences, are identified. *Ongoing communication negotiation* avoids the need for fundamental revisions of either behavior models or communication acts, as both are specified

simultaneously. Different viewpoints are immediately visible and can be discussed ad-hoc. This immediacy, at the same time, can be challenging for modelers, as they are continuously confronted with incoming messages or message requests while at the same time describing their own behavior.

The three approaches are described here without any preference and are currently subject to closer examination with regards to their practical applicability. The first experiences gained from these explorations are described in the next section.

Initial Experiences and Future Work

The modeling concept so far has been deployed in all three methodological variations in two different practical settings. In all cases, the models were built using paper-based modeling cards without any technical modeling support. In case A, the process of assembling a pneumatic cylinder was subject of elicitation. The actors were 8 students (6 male, 2 female) of business information systems, who were trained in the production process for 2.5 days as part of a practical course. All students already had extensive experiences in business process modeling. The process involved four different roles, of which each was taken by two students. All three methodological approaches were conducted using three different but equally complex variations of the production process. Case B was taken from healthcare sector, where 6 healthcare professionals (4 female, 2 male) modeled the admission process of an elderly client to long-term care (involving 4 roles in total). None of them had prior experiences with modeling, neither were they confronted with explicit process models in their professional life. All three methodological approaches were used in different steps of the elicitation process.

Overall, all participants in either case were able to create correct models (in terms of how the modeling elements were used and linked to other model parts). Differences, however, occurred during the modeling process, which can be attributed to the different backgrounds and prior experiences in modeling. Modelers with no experiences in modeling (in case B) repeatedly showed problems in understanding or correctly using the message elements. They were not able to consistently distinguish between the act of sending or receiving a message and the message itself and consequently had problems in assigning designators to messages. This problem was less evident in ex-ante communication negotiation, where people were not introduced to sending and receiving activities. The misconceptions could largely be overcome by providing examples of correctly designated messages.

The unexperienced modelers in case B preferred ex-post communication negotiation over the other two variants. They were unable to handle the complexity of the ongoing message negotiation setting and did not manage to incorporate the incoming messages while modeling their own role's behavior. Ex-ante communication specification was perceived to cause superficial effort, as they felt they had to go through their work process twice to first identify their communication and then to actually create the model. Two modelers also felt constrained by the already existing messages in modeling their role's behavior. The ability to uncover inconsistent

communication expectations in ex-post communication negotiation was repeatedly noted as the most useful aspect of the whole modeling session.

Preferences were different for the experienced modelers in case A. They preferred ongoing message negotiation as it was perceived most efficient and being the fastest way to reach a consistent overall model. Ex-ante message negotiation was considered a well suited approach and hardly led to the need for revisions after behavior modeling. It was generally preferred over ex-post message negotiation, which was considered to be cumbersome and hardly providing any added value.

As no clear preference for any methodological approach can be identified, future research will further examine the effects of the different points in time during modeling when communication acts are made explicit. The next steps are to resolve issues in understanding the semantics of the modeling elements — in particular the message element — for inexperienced modelers. In a further step, advantages and disadvantages of the different methodological approaches for different combinations of modeling goals, and prior modeling experiences of the participants will be examined. A more elaborate empirical setting will be used to overcome the limitations of the explorations described above, which mainly suffer from limited comparability and observably different complexity of the modeling subjects (with the production process being more accurately describable than the healthcare process).

A second strain of research and development currently worked on is tool support. Based upon an existing interactive tabletop modeling environment (Oppl and Stary, 2011), means to support a technically augmented distributed modeling workflow, including the currently omitted requirements on communication support, are currently implemented. Additionally, a mapping between the the modeling language used here and S-BPM (Fleischmann et al., 2012) is currently worked on, which will allow to validate and execute the created models. This will enable to identify model errors that can remain undiscovered due to the distributed nature of model elicitation (such as dead-locks by mutually waiting for messages to be sent).

Summarizing, the approach presented here is a first step towards business process model elicitation that explicitly uncovers existing or potential collaboration issues in organizational work processes. The modeling approach enables to independently create models for each involved role and aligning the communication acts among these roles in the course of the modeling act. The resulting models can directly be mapped to modeling languages that are supported by BPM tools for validation or workflow support. While the initial experiences with applying the modeling approach in practical settings are promising, future research has to adress methodological considerations to facilitate model alignment across roles as well as technical means of modeling support.

Acknowledgments

The research leading to these results has received funding from the European Commission within the Marie Curie Industry and Academia Partnerships & Pathways (IAPP) programme under grant agreement n° 286083 (IANES).

References

- Adamides, E. D. and N. Karacapilidis (2006): ‘A knowledge centred framework for collaborative business process modelling’. *Business Process Management Journal*, vol. 12, no. 5, pp. 557–575.
- Antunes, P., D. Simões, L. Carriço, and J. A. Pino (2013): ‘An end-user approach to business process modeling’. *Journal of Network and Computer Applications*.
- Dann, H. D. (1992): ‘Variation von Lege-Strukturen zur Wissensrepräsentation’. In: B. Scheele (ed.): *Struktur-Lege-Verfahren als Dialog-Konsens-Methodik*. Aschendorff, pp. 2–41.
- de Cesare, S. and A. Serrano (2006): ‘Collaborative Modeling Using UML and Business Process Simulation’. In: *Proceedings of HICSS’06*. IEEE Press.
- Fleischmann, A., W. Schmidt, C. Stary, S. Obermeier, and E. Börger (2012): *Subject-Oriented Business Process Management*. Springer.
- Gasson, S. (2005): ‘The Dynamics of Sensemaking, Knowledge, and Expertise in Collaborative, Boundary-Spanning Design’. *Journal of Computer-Mediated Communication*, vol. 10, no. 4, pp. 00–00.
- Giaglis, G. M. R. (2001): ‘A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques’. *International Journal of Flexible Manufacturing Systems*, vol. 13, no. 2, pp. 209–228.
- Herrmann, T., M. Hoffmann, G. Kunau, and K. U. Loser (2002): ‘Modelling Cooperative Work: Chances and Risks of Structuring’. In: *Cooperative Systems Design, A Challenge of the Mobility Age. Proceedings of COOP 2002*. pp. 53–70, IOS press.
- Herrmann, T., K. U. Loser, and I. Jahnke (2007): ‘Sociotechnical walkthrough: a means for knowledge integration’. *The Learning Organization*, vol. 14, no. 5, pp. 450–464.
- Oppl, S. and C. Stary (2011): ‘Effects of a Tabletop Interface on the Co-Construction of Concept Maps’. In: *Proceedings of the 13th IFIP TC13 Conference on Human-Computer Interaction (INTERACT 2011)*. pp. 443–460, Springer.
- Rittgen, P. (2007): ‘Negotiating Models’. In: J. Krogstie, A. Opdahl, and G. Sindre (eds.): *Advanced Information Systems Engineering*. Springer Berlin / Heidelberg, pp. 561–573.
- Rittgen, P. (2010): ‘Success factors of e-collaboration in business process modeling’. In: *Advanced Information Systems Engineering*. pp. 24–37.
- Soderstrom, E., B. Andersson, P. Johannesson, E. Perjons, and B. Wangler (2002): ‘Towards a framework for comparing process modelling languages’. In: *Proceedings of the 14th International Conference on Advanced Information Systems Engineering, CAiSE*. pp. 600–611, Springer.
- Stuit, M., H. Wortmann, N. Szirbik, and J. Roodenburg (2011): ‘Multi-view interaction modelling of human collaboration processes: a business process study of head and neck cancer care in a Dutch academic hospital’. *Journal of Biomedical Informatics*.
- White, S. A. (2004): ‘Introduction to BPMN’. *BPTrends*.

Operationalizing Dialogue Games for Collaborative Modeling

Stijn Hoppenbrouwers^{1,2}, Rob Thijssen¹ and Jan Vogels¹

¹Radboud University Nijmegen, the Netherlands

²HAN University of Applied Sciences, Arnhem, the Netherlands

stijn.hoppenbrouwers@han.nl,

jan.h.vogels@gmail.com,

thijssen_rob@hotmail.com

Abstract. In our ongoing attempt to create and support workable “rules of play” for effective, goal-driven conversations in the context of collaborative modeling, we have now reached a stage in which a first operational setup for such ‘dialogue games’ has been developed and tested. In this paper we present the basic, generic structure of a game-like procedure description, based on a prototype game for conceptual (information) modeling. The procedure has been tested and refined in a modest design cycle and has been demonstrated to work realistically. Its generic structure is meant to be applicable for different types of conceptual modeling, e.g. process modeling, goal modeling, task modeling, and so on (requiring specific rules and configurations for the different types). However, in this paper we merely aim to present our basic approach. An illustrative example is provided.

Dialogue Games

Dialogue Games are an innovative and, admittedly, experimental interaction form meant to support, structure and guide conversations for modeling. This functionality is desirable in context of helping people (in particular, novice analysts and modelers) to efficiently and effectively proceed in a goal-driven, focused collaborative effort to create a rational conceptualization or model (Hoppenbrouwers & Rouwette, 2012). Building on previous work, we present the outline of a workable set-up for dialogue games that can be used as a basis for a

set of such games, to be used in education and possibly also in practice in fields like requirements modeling, business modeling, process modeling, etc.

Dialogue games originated as a theoretical concept from argumentation theory going back to Wittgenstein's "language games". More recently they have been operationalized in the form of structured chats (Ravenscroft & McAlister, 2006) using the simple mechanism of *openers*: offering a small set of preset text fragments from which chat participants can choose to start a chat entry. For example, basic openers could be: "I have a question: ...", or "That is a bad idea because ...".

These openers guide participants in structuring the chat, and also make this structure visible. They create clear, natural structure in the log resulting from the chat. (Hoppenbrouwers & Rouwette, 2012) first employed this idea in context of collaborative modeling. The underlying concept here is that every collaborative modeling effort essentially is a conversation producing that model (Ssebuggwawo, Hoppenbrouwers, & Proper, 2009). As the conversation progresses, the model advances, every step being traceable in the log. The chat includes not only propositions for additions or changes to the developing model, but also the extensive conversation *about* the propositions: questions, answers, argumentation, negotiation.

(Hoppenbrouwers & Rouwette, 2012) showed that the opener mechanism can be extended to include semantic and syntactic structuring besides discourse structuring, by means of providing not only openers but also 'fill-in' patterns for answers. For example, as an answer to the (predefined) question "Please give the actor role responsible for executing this activity", the predefined answer structures offered in that context could be:

- "The actor role responsible for activity [ACT] is [AR]"
- "Sorry, I cannot answer that question"

By making the structured chat opener mechanisms context sensitive, it is possible to purposefully shift conceptual and conversational focus, dynamically entering consecutive Focused Conceptualization modes or 'FoCons' (Hoppenbrouwers & Wilmont, 2010). FoCons represent brief spans in a conversation in which a clear and limited focus is maintained, for example "list all roles for process P and their related activities". Which focus to choose at what point in the conversation is typically decided by a facilitator. We offer a workable, game-like structure ("rules of play") providing guidance and support in such conversations-for-modelling, to both the facilitator (primarily asking questions) and one or more participants (mostly doing the answering). The spirit of this setup is not unlike that of a semi-structured interview, but with more explicit, model-oriented goals and structure. By also providing some generic conversational openers throughout the game, it is possible to blend highly focused conversation with more generic questioning and answering. Also, it is possible to combine the use of dialogue games with existing diagramming tools

(Hoppenbrouwers & Rouwette, 2012), for example SeeMe (Herrmann, 2009). Finally, the setup is meant to be combined with practices and principles from group facilitation and collaboration engineering (Hoppenbrouwers & van Stokkum, 2012).

Below we will briefly present the main aspects of a prototype implementation of our dialogue game concept. The game outline presented is an integration of two related projects in which prototypes were developed cyclically (testing leading to improvements) and finally evaluated through more tests, leading to observations, remarks from test subjects, and their completing brief questionnaires addressing matters like understandability and playability of the game, perception of purpose and usefulness, and participant experience. For more on this, we refer to the original master's theses: (Vogels, 2013) and (Thijssen, 2013).

Game Structure and Rules

The dialogue game sketched in this section aims at the creation of a specific dialogue game for the conceptualization phase of model elicitation following the FCO-IM method for conceptual information modeling (Bakema, Zwart, & van der Lek, 2002). For our current purpose this method should be seen as a case and illustration. It was chosen because its operationalization is a main interest in our research group, and because it includes a well described elicitation procedure. We focus mainly on the generic setup, which should be useful for a broad range of modeling or conceptualization games. The illustrative examples, however, still concern basic FCO-IM conceptual modeling, which is a form of fact-oriented data modeling.

The game distinguishes between two roles: the **facilitator**, typically a skilled modeler/analyst who initiates and leads the game, and one or more **participants**, typically people with good domain knowledge but not necessarily with any background in systems thinking or modeling. In our prototype, we work with only one participant. The game as presented thus supports a collaboration between facilitator and participant only. However, by adding a multi-participant layer to the game, it is quite simple to extend the dialogue to a collaboration *between multiple participants* (as well as the facilitator): they propose, discuss and improve an answer before (and after) it is entered as a statement in the main dialogue. This mechanism has been successfully explored and demonstrated in (Hoppenbrouwers & Rouwette, 2012).

The top node in the hierarchical decomposition of our game procedure is the **game** as a whole. One focused modeling session will typically be done by playing the game once. The game is divided into **phases**. A phase represents a distinct part of the modeling procedure, with a distinct conceptualization goal. For example, in our prototype we split the FCO-IM elicitation procedure into two.

The example game covers only the first phase (“conceptualization”): identifying and explicitly formulating the domain concepts and relations (words, phrases) sought for, within the conceptual frame (meta-model) of the FCO-IM method. The second phase, discarded in this paper, concerns the systematic elicitation of *constraints* on populations of the model. The FCO-IM procedure requires the gathering of example sentences (‘facts’) used in recurrent communication patterns in the domain. These examples are then rephrased at a higher level of abstraction (thus going from instance level to ‘fact type’ level). The elementary type-level phrases so created are the basis for a formal structure (model) that can also be represented diagrammatically, showing the related concepts and relations (language elements) used in the domain: its ‘information grammar’ (Bakema et al., 2002).

Covering a phase iteration from start to end is called a **round**; a round is one instance of the activities belonging to a phase. In our example, an attempt at fully charting one phrase (objects and the relation(s) between them), covering one element of the domain description (called a fact type), constitutes one round. The structure of the phase allows for some freedom of action within a round, so rounds do not have to be played out identical to one another. The end of a round is a decision whether to start a new round (because the result is still incomplete or otherwise not satisfactory) or end the phase. The rules clearly describe how to proceed at this point to force a decision about ending the round. Importantly, at any point in the game the facilitator may decide to revisit completed concepts for amendments. These *ad hoc* flow decisions can override the standard flow.

Each phase of the game is subdivided in a number of **steps**. Steps are small separate sets of related activities within a round that are more operationally described than phases and rounds. Going through one iteration of a step is called a **turn**. Multiple turns can be taken to finish a step.

A **mission list** is created at the start of the game and continuously updated throughout the game. It plays a vital part in keeping focus and helps the facilitator make decisions. The mission list consists of a top level goal (the main goal of the session) and sub-goals. The facilitator has full control over the mission list. For each concept which emerges during elicitation, a number of sub-goals are created and then pursued, systematically asking questions related to the concept (based on the method’s meta-model). The current conceptualization goal is highlighted and finished goals are crossed off. The mission list (or a summary thereof) can also be viewed by the participant. Below we show an example of the mission list in mid game. Strikethrough items are goals accomplished; the highlighted concept is the one focused on in the current round. The various sub-goals are achieved through the various FoCons in the steps of the round. Space prevents us from explaining the method specific modeling terms used in the example; for more on this, see (Bakema et al., 2002).

- Create FCO-IM model of “student registration” domain
 - ~~Concept “Course” [+]~~
 - Concept “Student” [-]
 - ~~Generate 4 examples of “Student”~~
 - ~~Generate elementary fact~~
 - Elicit identifier
 - Generate LTL-FTE for repository
 - Generate OTL-FTE for repository
 - OPTIONAL: identify uniqueness constraint (UC)
 - OPTIONAL: identify totality constraint (TC)
 - Draw part of the Information Grammar Diagram
 - Validate drawn Information Grammar Diagram and repository information

The flow of the game is directed by **decisions** made by the game facilitator. Some decisions can be made by applying clear-cut, discretely decidable rules, like “if only one unfinished concept goal remains on the mission list, select this goal and move to step 2.” Sometimes, however, a more intelligent involvement of the facilitator is required. For example, it can be decided to skip a repeated and possibly annoying validation step, at some risk (for example: “is this phrase correct in the domain: <student> follows <course>”). In such cases, the rules may offer suggestions (“only skip a validation if you feel fully confident that further validation is not necessary”) but leave the decision to the facilitator.

The standard activities of which a step consists are: **FoCon**, **game procedure** and **decision**. FoCon dialogue rules to a certain degree regulate and support the actual facilitator-participant interaction within the game: the focused question-and-answer part. We divided the prototype phase into three steps based on three FoCons identified in the original FCO-IM procedure: “Generate Concepts”, “Qualify Concepts”, and “Validate Information Grammar Diagram (IGD)”.

‘Game procedures’ concern administrative background work done by the facilitator, that in advanced digital implementations of the game could be partly or fully automated. For example, the updating of the mission list is such game procedure, as is the drawing of an IGD. Decisions concern the flow of the game as explained above.

The components as discussed shape the flow of the game, which is unpredictable (within limits). The **rules** specify the details, as explained above. Flexibility is important to deal with unexpected situations and allow for ad hoc iterations. We provide defaults which can be followed by novice information analysts who are unsure how to proceed or react. With this mechanism, we avoid the bogging down of the procedure. The flow rules guide the facilitator. The only flow rule for the domain expert is: “follow the lead of the facilitator”.

Questions and answers are at the heart of the game. The facilitator will try to elicit the content by systematically (but not too rigidly!) asking questions to the participant. The questions structure the conversational part of the game. Each FoCon comes with a dedicated repertoire of question and answer templates. As explained in the first section, some of these are highly focused, others very generic. The facilitator may give an example answer to help the domain expert conceive and formulate a correctly formed answer (S. Hoppenbrouwers, 2012). It is not mandatory to use every question available in a turn. Some examples of predefined questions in step 2 (phase 1) of the game are the following:

- Could you give a meaningful name for this <object type>? For instance, Fido is a *Dog*; Mercedes Benz is a *Car Brand*. (Elicits a meaningful type name for an object, label or fact type)
- How are <object>s identified? For example, a ‘Dutch Citizen’ has a name but also a unique Citizen Service Number (Elicits an identifier for a concept)
- How do you distinguish between <object>s in your communication? (Auxiliary question for eliciting an identifier for a concept)
- Can there be two <object>s with the same <identifier>? (Validates the uniqueness of an identifier)

The question and answer mechanism can be implemented using some basic software. Main inspiration and example for such software is the Interloc system for generic dialogue games (Ravenscroft & McAlister, 2006). Central to this enhanced chat system is the use of ‘openers’, as explained in the first section. Additional rules are the following.

1. A participant and facilitator alike can only speak when it is their turn to speak (i.e. turn taking is enforced). Experiences with Interloc have confirmed that this is a highly desired feature in a structured chat.

2. Relevant categorized elements (words, phrases) in the answers are marked (in the prototype: by hand, by the facilitator) so they can be easily recognized and traced by both human players and the system. Such items, once identified, can be added to the mission list, planning further question asking about them. As mentioned, the items are typically related to the meta-model of the modeling language underlying the game, but may also represent intermediate concepts (‘half products’) in the process towards a target conceptualization. For example, an elicited instance-level object (“John Doe”) will still have to be categorized and named, and will be used as input for the elicitation of an object type concept (“Student”).

3. In addition to offering a dedicated repertoire of question and answer patterns for each FoCon, a context-sensitive mechanism is added that limits the availability of questions and answers for every consecutive chat entry within the FoCon (i.e. each conversational **move**). This mechanism is role differentiated, meaning that depending on the context (step, previous move) a move within the

FoCon (for example, asking one question or giving one answer) is offered only to either the facilitator or the participant. Through the mechanism, ‘question flow’ is constrained, which turned out to work nicely. However, the flow rules can also be left void, allowing an unrestricted open FoCon flow based on a set of available questions and answers without a constrained temporal ordering or role-specific availability.

4. The interface for choosing openers first offers a choice between some briefly described question or answer options. Only after a choice is made, a matching *text template* is offered which can be altered at will by the player. This includes the tagging of specific items that make them machine recognizable. For example, at a particular point in step 2, one of the options for a facilitator move is “Ask for distinctive object identifier”. Choosing this option provides the template (already listed above) “How do you distinguish between <object>s in your communication?”. The facilitator can then fill in the <object> slot, effectively entering the move “How do you distinguish between distinguish Students in your communication?” in the FoCon dialogue. Next, one of the moves offered to the Participant is “Give the identifier asked for”. If chosen, the template plus example offered are “<object>s are distinguished by (For example: Cars are distinguished by Licence Numbers).

In an advanced version, it should be possible to make some parts of the template freely adjustable, and some not adjustable. This can be augmented by partially automated tagging.

Conclusion and future directions

The synthesis of two prototypes, also extending previous and less guided experiments (Hoppenbrouwers & Rouwette, 2012), have served well in creating operational and clearly structured setups for advanced, specialized collaborative dialogue games. The next step will be to create a robust, user friendly and generically usable digital environment combining the best features of the various prototypes. As in any software development project, there will no doubt be new challenges and insights as we continue our effort to provide user friendly support for guided/structured conversations in collaborative modeling. In addition, we aim to include links with visualizations and verbalizations mirroring the concepts put forward in the conversation, thus completing the connection with more traditional, diagram-based forms of modeling. Finally, we continue our effort to include ‘groupware’ and group facilitation features in our dialogue games.

References

- Bakema, G., Zwart, J. P., & van der Lek, H. (2002). *Fully Communication Oriented Information Modeling (FCO-IM)*. Arnhem: FCO-IM Consultancy.
- Herrmann, T. (2009). Systems Design with the Socio-Technical Walkthrough. In B. Whitworth & A. d. Moor (Eds.), *Handbook of research on Socio-Technical Design and Social Networking Systems* (pp. 336-351). Hershey: Idea Group Publishing.
- Hoppenbrouwers, S. (2012). *Asking Questions about Asking Questions in Collaborative Enterprise Modelling*. Paper presented at the The Practice of Enterprise Modeling, 5th IFIP WG 8.1 Working Conference PoEM 2012.
- Hoppenbrouwers, S., & Rouwette, E. (2012). A Dialogue Game for Analysing Group Model Building: Framing Collaborative Modelling and its Facilitation. *International Journal of Organisational Design and Engineering, special issue on Collaborative Modeling*, 2(1), 19-40.
- Hoppenbrouwers, S., & Wilmont, I. (2010). Focused Conceptualisation: Framing Questioning and Answering in Model-Oriented Dialogue Games. In P. Bommel, S. Hoppenbrouwers, S. Overbeek, E. Proper & J. Barjis (Eds.), *The Practice of Enterprise Modeling* (pp. 190-204). Berlin, Heidelberg: Springer.
- Hoppenbrouwers, S. J. B. A., & van Stokkum, W. (2012). From Dialogue Games to m--ThinkLets: Overview and Synthesis of a Collaborative Modeling Approach. *International Journal of E-Collaboration (IJeC), special issue on Collaborative Usage and Development of Models and other Visualizations*.
- Ravenscroft, A., & McAlister, S. (2006). Designing interaction as a dialogue game: Linking social and conceptual dimensions of the learning process. In C. Juwah (Ed.), *Interactions in Online Education: implications for theory and practice* (pp. 73-90). New York: Routledge.
- Ssebuggwawo, D., Hoppenbrouwers, S., & Proper, E. (2009). Interactions, Goals and Rules in a Collaborative Modelling Session. In A. Persson & J. Stirna (Eds.), *2nd IFIP WG8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2009)*. Berlin, Heidelberg: Springer.
- Thijssen, R. (2013). *An argument based approach for easier process modeling using dialog games*. Radboud University Nijmegen, Nijmegen.
- Vogels, J. (2013). *A serious gaming approach to content elicitation for FCO-IM*. Radboud University Nijmegen, Nijmegen.

Beyond Collaborative Model Usage and Development – A Model Lifecycle Approach for Lay User Modeling

Alexander Nolte, Michael Prilla

Information and Technology Management, Institute for Applied Work Science,
Ruhr University of Bochum; *{nolte|prilla}@iaw.rub.de*

Abstract. Approaches of collaborative modeling and model usage aim to increase the participation of stakeholders in modeling, but either still require experts support or are bound to certain phases of the model lifecycle: This makes it hard to compose an overall concept of collaborative model usage and development. In this paper, we argue that we need concepts to engage users without modeling capabilities into self-directed, user-managed processes of using and working on models. We present a corresponding model lifecycle as well as suitable interaction and participation modes, using examples from our ongoing work on integrating lay-users into model usage and development. We analyze this work and present open issues to be discussed by the community.

Introduction: Cooperation beyond Participatory Model Usage and Development

Process models are common tools in modern organization. Most approaches of using them for analysis, specification and guidance in organizations have been developed and designed for expert users, that is, people trained in process analysis, modeling or model usage. Recent work has amended that this does not involve stakeholders in a way that encourages them to become active model users themselves (e.g., Hoppenbrouwers et al. 2010; Prilla and Nolte 2012; Rittgen

2010). This in turn also potentially leads to diminished commitment, motivation and agreement to processes. Furthermore expert support is costly and delays model development (Nolte & Prilla, 2013; Rittgen, 2010). Approaches for collaborative model usage and development consequently have emerged as research fields in recent years. While some of these approaches are at supporting collaborative modeling by experts, others explicitly integrate process stakeholders.

The (ongoing) work we present here aims at taking these approaches one step further, towards the support of self-directed, user-managed collaborative usage and development, which can be performed by users without expertise in modeling as most collaborative modeling solutions still rely on expert support (Rittgen, 2010) and some also limit participants to verbal contributions (Herrmann, 2009). This reduces stakeholder involvement as e.g. when a model has reached a stage where it serves as a source for software development, stakeholders are usually cut from the possibility to give feedback if changes occur or to suggest changes if they make experiences in practice that afford them. We argue that **stakeholders need to be integrated into model development and usage throughout the entire model lifecycle** (see Nolte and Prilla 2013 for a detailed discussion). They also have to be integrated more tightly thus requiring a concepts and corresponding tool design to enable them to be active in corresponding tasks, even if (or especially in a case when) they are not modeling experts. In this paper, we present a formalization of these tasks and results of ongoing work in supporting them.

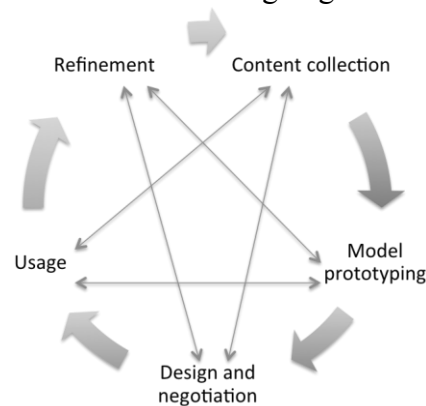


Figure 1: A model lifecycle for collaborative usage and development of models.

The Model Lifecycle

There are a lot of approaches and tools for collaborative modeling and model usage that include participants other than modeling experts. They vary from approaches requiring (expert) facilitation to self-directed modeling and model usage (Nolte & Prilla, 2013). Despite this work, in practice models are often only used by experts. We argue that one of the reasons for this is that existing support is not well aligned to a model lifecycle that integrates stakeholders into model usage and development – only if self-directed modeling throughout all its phases is support-

ed, it has a chance to become an established practice in organizations. Based on and inspired by existing literature (e.g., van der Aalst et al. 2003; Dumas et al. 2013; Prilla et al. 2013; Rittgen 2010) as well as work done in our group (Herrmann, Nolte, & Prilla, 2013; Herrmann, 2009; Nolte & Prilla, 2013; Prilla & Nolte, 2012), we have derived a prototypical model lifecycle that is tailored to collaborative model usage and development, describing relevant phases in which participants can be actively integrated. Figure 1 shows this lifecycle and Table 1 gives a brief description of its phases. The sequence is not mandatory – phases might have to be conducted multiple times before arriving at a usable model.

Table 1: Phases in the participatory model lifecycle and existing approaches.

Phase	Description	Approaches
Content collection	As a preparation, experts, stakeholders and others loosely gather necessary information and content.	<i>Experts:</i> Interviews, document analysis (Dumas et al., 2013), Ethnography (Herrmann, 2009) <i>Participatory:</i> Ideation (Herrmann et al., 2013), Collection (Andersen & Richardson, 1997)
Model prototyping	The material available for modeling is examined, necessary material is chosen and an initial model prototype is created to work with.	<i>Experts:</i> Creation of initial model from material, process mining <i>Participatory:</i> Clustering (Wiechers, Nolte, Ksoll, Herrmann, & Kienle, 2013), structured conversation (Hoppenbrouwers et al., 2010)
Design and negotiation	Together with stakeholders, the model is designed and negotiated to represent a process that all participants agree on for implementation.	<i>Experts:</i> Face to face meetings, workshops, verbal / written feedback (van der Aalst et al., 2003) <i>Participatory:</i> Voting, facilitation (Herrmann, 2009), collaborative modeling (Rittgen, 2010)
Usage	The model is used by various stakeholders, e.g. developers implementing support or workers using models as guides	<i>Experts:</i> Model presentation, workflow engines <i>Participatory:</i> Models in wikis (Rospocher, Ghidini, Pammer, Serafini, & Lindstaedt, 2009), models as means of knowledge exchange (Cherubini, Venolia, DeLine, & Ko, 2007)
Refinement	According to experiences from using the model or process, the model is refined.	<i>Experts:</i> Measurements, e.g. KPI (Weske, 2007), feedback <i>Participatory:</i> Critiquing (Herrmann et al., 2013), Walkthrough, Commenting

While the lifecycle shown in Figure 1 is not only applicable to self-directed modeling, but also to modeling procedures guided or solely conducted by experts (as Table 1 shows), its structure enables us to assess the state of the art in support for

collaborative model development and usage and also to describe challenges in enabling stakeholders to actively develop and use models in a self-directed way:

Content collection: An important part of model development is gathering information about the process. This includes process steps (activities) as well as roles carrying them out and resources used by them. Table 1 shows a variety of expert-driven approaches methods and tools supporting this phase, while there is a lack of self-directed, user-managed approaches. Non-expert users need to be supported to gather model content – some expert-driven modeling approaches even skip this phase, merging it with the following one.

Model prototyping: Building on the content collected before, this phase is intended to create a process shape and to allocate the content (activity steps, actors etc.) to it, thus aiming at creating a first representation of the process. This proves to be challenging for people without modeling expertise, as it requires a translation from mental models to model language. While most approaches supporting this phase rely on expert support, some allow for self-directed alignment of process content through e.g. clustering. This cannot be expected to result in a full-fledged process model it is a first step in model prototyping. The challenge thus is how interaction with models and modeling tools can be designed to allow users without deep modeling knowledge to create a useful model prototype.

Design and negotiation: This phase aims at creating a process model out of the previously developed prototype that all participants can agree on and that includes the necessary details for implementation (either within the organization or by support of tools). This process might be difficult, as differing views about certain process steps may be present that have to be negotiated and represented in the model. Therefore, most approaches supporting this phase use some kind of expert facilitation. There are however approaches such as voting that may serve as a support for negotiation and allow for participants to become actively involved.

Usage: After completing the model different stakeholders (e.g., workers, management, developers) can use it to guide work, transfer knowledge or use it as a reference for tool implementation. This usage may raise questions about the content or details of the model and it might impose high cognitive and time efforts because of the complexity that models may have. As there is not always a facilitator present to describe the model in an adequate abstraction or to answer questions on details, in self-directed modeling The challenge is to enable people to work with models without this support.

Refinement: Similar to BPM lifecycles, the refinement phase of model usage and development aims at integrating experiences from practice and measurements taken on the performance of the process into a process model thus revising and improving it. While there are approaches that relate lacking performance to steps in models and thus allow focused improvement of these parts, for more informal feedback of stakeholders or self-directed reflection of processes currently hardly any approaches are available. If modeling is to be done by users in a self-sustained

manner, this phase needs to be supported, as otherwise models will soon either become outdated or will show an idealized view instead of real work processes.

Regarding these challenges, the questions arise (1) whether we can support these phases of model development and usage in a way that enables self-directed model interaction for stakeholders others than experts and (2) how support for these phases can be created and smoothly connected to the respective other phases in order to support the whole lifecycle. In what follows, we relate our work to the lifecycle and identify open issues and questions to be tackled by future research.

Support for Lay User Modeling

Based upon the previously described model lifecycle we will now describe our approaches to integrate lay users into them. Besides these proposals, future research needs to clarify the role of expert facilitation in the phases.

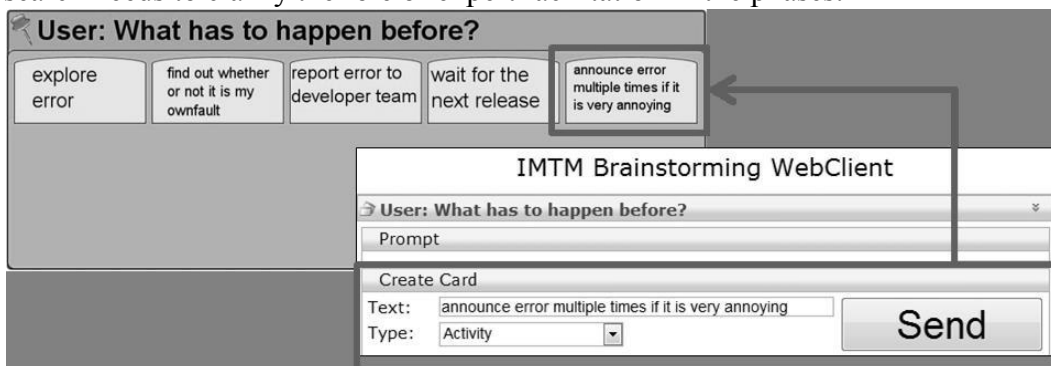


Figure 2: Written text in a web interface (bottom) resulting in an element with the respective label within a model (top).

Content collection: Figure 2 shows a system that transfers written text into elements of a modeling notation with the respective text as their label. It allows for non-modeling experts to contribute content to a model by either adding content on to an initially empty model or using pre-defined model parts as target areas for contributions as shown in Figure 2. This enables the collection of content into an existing model and it also allows for pre-structuring the collection by providing areas covering different aspects of a process such as activities conducted or resources required. While the latter approach has proven to be feasible in a workshop setting, it requires preparation by experts defining the aforementioned areas. In order to improve this we came up with the idea to guide participants through content collection by sequentially asking them predefined questions such as “What happens next?” or “Who does that and which resources are required?” thus mimicking a walkthrough approach (Nolte & Prilla, 2013). This system however is still in a prototypical state and has not been tested yet.

Model prototyping: Creating a process model based upon loose contributions requires in-depth knowledge of a modeling notation. Non-modeling experts can however certainly prepare this step as they are capable of aligning activities with

respect to their position within a process sequence and also assign roles to the respective activities that are involved in conducting these activities (Prilla & Nolte, 2012). In order to support this we developed a mechanism that allows users to align elements within clusters (c.f. Figure 3 right) by simply touching them and dropping it at the designated position (Wiechers et al., 2013). This is an initial step to support model prototyping, but important process characteristics such as conditions and flows are still missing.



Figure 3: Screened model elements that have been marked (left) and are put into clusters collaboratively afterwards (right).

Design and negotiation: This phase is about forming model that depicts the process in an accurate manner and that all participants agree on. In an expert-driven approach, we would typically support this by inviting all relevant process stakeholders to a workshop where a facilitator sequentially walks them through the process the model depicts. In order to involve participants even more actively we developed a mechanism that allows users to mark elements (see the circles in Figure 3 left) by touching them. While this mechanism allows non-modeling experts to directly interact with the model, this phase is still dependent on the facilitator and also requires all stakeholders to be present at the same time. This led us to the idea of building a system that prompts users for modeling actions by asking questions similar to the ones described before in the content collection phase (Nolte & Prilla, 2013).

Usage: In order to use models for work that is e.g. to gather information about a process, people have to have access to it. While this sounds trivial at first sight it is far from being a common practice in organizations: usually only process owners or corporate process management have access to them. Also the software that is used to view a process is often complex to use for people not trained in using it (because it is built for the need of modeling experts). Furthermore, for a model to be useable it has to be presented in a way that non-modeling experts can make sense of it on their own. While we support access through an easy to use web based system that does not require any additional software to be installed (Figure 4), presenting it in a suitable manner still remains an issue. While the system allows for steps of a process to be hidden and later be shown again to the user (thus supporting exploration), it is still very static.

Refinement: As this phase is tightly connected to the previous one due to the necessity of using models in order to being able to refine them, our means of supporting this are very similar. The aforementioned web based system not only al-

allows for displaying and exploring models but also provides the opportunity to attach textual comments to any element of the model (c.f. Figure 4). Combining a familiar means of articulation (textual input) with access to the models (see above) allows process stakeholders to reflect on their processes during every day work – the comments they leave are later included in the model. While this is a rather simple solution, it keeps the usage barrier low and allows people that usually are cut from further model development to become pro-active model users. Furthermore the content and number of comments can provide process management departments with useful information about whether processes are up to date or need further refinement. Feeding back the comments into the model then usually happens within modeling workshops.

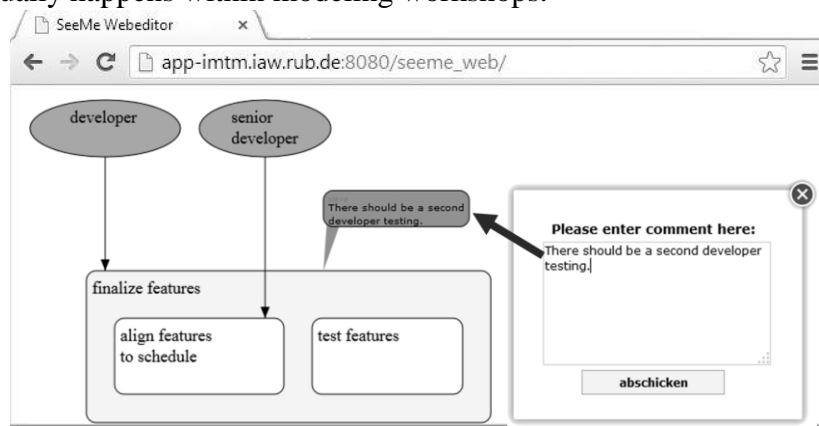


Figure 4: A textual comment that is attached to a model element.

Solutions and open issues

We presented a participatory model lifecycle and its respective phases. We also showed current support and issues for self-directed non-modeling expert participation within these phases. While it became apparent that phases like *content collection* and *model prototyping* are supported in a promising way, others still lack proper support for process stakeholders to become active in model development and usage. Especially phases like *design and negotiation* and *usage* still rely on experts. For *design and negotiation* we are currently planning to make use of milestones as scaffolds thus supporting process rather than content related clustering. Using approaches such as Kanban and Gantt diagrams for project planning that are known to process participants might also be beneficial. Our future work will focus on discussing current issues thus aiming at enhancing existing support for non-modeling experts developing and using models throughout the model lifecycle. Furthermore we are aiming at seamlessly intertwining these phases thus creating an approach that ties them closer together.

References

- Andersen, and Richardson, (1997): 'Scripts for group model building', *System Dynamics Review*, 13 2, pp. 107–129.
- Cherubini, Venolia, DeLine, and Ko, (2007): 'Let's go to the whiteboard: how and why software developers use drawings', *Proceedings of the SIGCHI conference on Human factors in computing systems* pp. 557–566.
- Dumas, La Rosa, Mendling, and Reijers, (2013): *Fundamentals of Business Process Management*, Springer.
- Herrmann, (2009): 'Systems Design with the Socio-Technical Walkthrough', In B. Whitworth & A. de Moor (Eds.), pp. 336–351 Information Science Publishing.
- Herrmann, Nolte, and Prilla, (2013): 'Awareness support for combining individual and collaborative process design in co-located meetings', *Computer Supported Cooperative Work (CSCW)*, 22 2, pp. 241–270. doi:10.1007/s10606-012-9179-x
- Hoppenbrouwers, Schotten, and Lucas, (2010): 'Towards Games for Knowledge Acquisition and Modeling', *International Journal of Gaming and Computer-Mediated Simulations, Special issue on AI and Games*, 2 4, pp. 48–66.
- Nolte, and Prilla, (2013): 'Anyone can use models: Potentials, requirements and support for non-expert model interaction', *International Journal of e-Collaboration. Special issue on "Collaborative usage and development of models."*
- Prilla, and Nolte, (2012): 'Normal users cooperating on process models: Is it possible at all?', *Proceedings of the 18th CRIWG Conference on Collaboration and Technology*.
- Prilla, Nolte, Herrmann, Kolfshoten, and Lukosch, (2013): 'Collaborative Usage and Development of Models: State of the Art, Challenges and Opportunities', *International Journal for e-Collaboration*.
- Rittgen, (2010): 'Collaborative Modeling: Roles, Activities and Team Organization', *International Journal of Information System Modeling and Design (IJISMD)*, 1 3, pp. 1–19.
- Rospocher, Ghidini, Pammer, Serafini, and Lindstaedt, (2009): 'MoKi: the modelling wiki', *Proceedings of the Forth Semantic Wiki Workshop (SemWiki 2009), co-located with 6th European Semantic Web Conference (ESWC 2009)* Vol. 464, pp. 113–128.
- Van der Aalst, ter Hofstede, and Weske, (2003): 'Business process management: A survey', *Proceedings of the 1st International Conference on Business Process Management (BPM)* pp. 1–12 Springer.
- Weske, (2007): *Business Process Management: Concepts, Languages, Architectures*, Berlin: Springer-Verlag.
- Wiechers, Nolte, Ksoll, Herrmann, and Kienle, (2013): 'User Tracking for Collaboration on Interactive Wall-Sized Displays', *Interaktive Kulturen 2013*.

The Added Value of Collaborative Modeling for Legal Business Rule Management

W. van Stokkum¹, P. Heiner¹,
S.J.B.A. Hoppenbrouwers² and H. Mulder³

¹Everest B.V., 's Hertogenbosch, the Netherlands

w.van.stokkum@everest.nl

p.heiner@everest.nl

²HAN University of Applied Sciences, Arnhem, the Netherlands

stijn.hoppenbrouwers@han.nl

³Antwerp Management School (AMS), Antwerp, Belgium

hans.mulder@viagroep.nl

Abstract. In this paper we discuss background considerations, domain properties, and some design principles for collaborative modeling environments combining the Business Rule Management approach and the Collaborative Modeling approach. The context focused on is that of translating law texts to operational processes and systems for implementing those laws in the public sector. The process of operationalizing law is very difficult to tackle: a diversity of stakeholders have to be involved to reach consensus on semantics, goals and business service design. We consider collaboration techniques crucial in order to create the required broad basis of acceptance regarding operational policies and their formulation. Collaboration techniques also enhance the efficiency and transparency of the process. We discuss the new role of collaboration in relation to the governance processes of the organizations. We illustrate a design case by describing an environment we are developing. We reflect on some lessons learned, concluding that adopting collaborative modeling techniques alone is not enough. Explorations show that additionally, rules and mechanisms are needed to structure and facilitate the group decision making process.

Introduction

This paper was written in context of an ongoing development project aiming to create a collaborative modeling environment developed to support Dutch governmental organizations in implementing legislation into their operational processes. Though we do sketch the current prototype environment and some design principles, the paper mostly concerns generic considerations about collaborative aspects in this application domain and its consequences for “law execution support systems”.

In the process of business rule creation and management, a variety of stakeholders (legal experts, business management, business architects, IT-experts) work together in order to translate legislation into usable specifications for operational business processes, including specifications for business rule driven IT-systems. In the Netherlands, as well as in various other countries with a thoroughly digitized governmental and public sector, such processes can be observed to exist in many domains (e.g. tax, customs, subsidies, permits, defense) and across a number of governmental organizations.

This process is commonly recognized as being very complex. Legislation is not directly usable in operational situations (typically, law execution by public service organizations). Terms and phrases used in legislation documents often contain pragmatic mismatches and contradictions because of the different contexts in which they are used.

Also, legislation tends to describe *WHAT* a policy should be, but not *HOW* it should be implemented by the variety of organizations that have to deal with it. Frequently the legislator deliberately leaves definitions and criteria vague, in order to let exact and definitive criteria arise in practice. In short: substantial additional policy making is needed to design business services and operational business rules that can be handled in everyday work processes.

The effects of ignoring collaboration

Current methods for ‘translating’ legislation into operational rules (though perhaps ‘developing’ would be a more accurate word here) typically have their origin in Business Rule Management practices. This discipline traditionally approaches the translation quite rationally, for example (Wyner, Engers, & Bahreini, 2010). The normal approach is to rewrite sources (legislation documents) directly into some sort of formal logic, in a format that can be logically validated and is suitable for further translation into executable rules that can be handled by business rule engines and other rule-based systems.

However, the actual translation process in practice can hardly be classified as being “rational” in the discrete and deterministic sense. Interpreting legislation and

the design of business services, thus implementing legislation, involves input of a variety of stakeholders. All stakeholders act according to their own perspectives and goals and use their own ‘domain specific language’. Traditional methods tend to ignore these aspects. They regard them as being the concern and responsibility of the “super IT-analyst”, who has to consult all stakeholders involved and unify their views and formulations. Such super analysts are very hard to find in real life, which causes a scalability issue within the organization. While the speed of implementation power increases dramatically due to the introduction of modern business process management platforms, analysis and design become the new bottlenecks (Hoppenbrouwers, Schotten, & Lucas, 2010).

Ignoring collaboration factors during the formulation of operational policies also introduces another serious issue. The business policies that will be implemented often only include the input of a limited set of stakeholders. In many cases, only a legal expert is consulted and legislation is rationally converted into some kind of logic. The resulting working instructions and systems often do not meet the views and practices of the knowledge workers that have to deal with real life cases. As a result, they feel unsafe because decisions are made that cannot be motivated or that do not take into account the situational context of cases in real life. In short, lack of a common base of understanding has negative effects like the leaving of valuable employees (not willing to adopt the policy made), customer unfriendly behavior (“the system is always right”; “computer says no”) or fraud and sabotage (manipulating real life data/facts in order to reach a desired outcome, or simply ignoring systems and policy), causing erroneous and inconsistent behavior of the organization’s services.

In order to fulfill the need for collaboration in law-based business rule creation and management, a modeling environment is being developed combining traditional business rule management techniques with those used in collaboration environments

Collaboration to support Shared Decision Making

Commonly known collaboration techniques (chat, shared annotation of documents, discussions, forum, mentioning, case management) are used to optimize the process of working together and making shared decisions. Collaborative techniques as in, for example, (de Vreede & Briggs, 2005; Kolfshoten, Briggs, de Vreede, Jacobs, & Appelman, 2006) turn out to be a crucial factor to tackle the issues in collaborative rule management experienced today

With the help of collaborative modeling techniques, a large group of people can be involved in shared decision making. By facilitating an online workspace, people will no longer be dependent on each other’s agendas. Asynchronous work reduces the need to physically meet during design and group decision making. A substantial larger number of people directly or indirectly can be involved.

When working in an environment deploying collaboration techniques, questions, answers and arguments in discussion are systematically logged so they can be shared and responded to at a later moment. This also allows for detection (not necessarily automatically!) whether people are arguing from the same perspective or not, which can help prevent misunderstandings and mistakes rooted in deviant interpretation.

Sharing knowledge is no longer limited to a point to point interchange between individuals. When reusing the model elements in multiple products and services, decisions and semantics will automatically be reused too. The possibility of reusing the outcome of the ‘translation process’, *including the underlying group conversation*, is essential for implementing consistent and correct behavior of organizational services. For the government, this means consistent and reliable behavior towards citizens and companies.

Computer supported collaboration support: a new enabler for compliance

In our modeling environment and domain, in addition to the common application of collaboration, computer supported collaboration did get another very important, and unexpected new role. It has been integrated with the governance processes of the organization.

The business rules (operational policies) created in the process are implemented in a business rule system. This system is able to automatically reason over these rules so it can automatically decide whether e.g. citizens or companies are entitled to receive subsidy or are allowed to receive a residential permit. Even the amount of tax citizens have to pay are calculated automatically. For the organization it is crucial to be able to explain why a decision is made. Although a direct link to relevant legislation sources at first hand seems sufficient, exact explanation can only be based on the design decisions made when formulating the operational policy. So when the system generates explanation it actively uses the arguments behind the discussions when formulating the business rules, in order to really provide 100% transparency in decision making. Without computer supported work this could never be realizable because these arguments never were systematically logged.

Lawyers use these outcome of collaboration processes results when judging official complaints being filed. They even use them when preparing the lawsuits.

A prototype environment for collaborative rule management

The prototype environment in development enables various stakeholders in developing multiple interconnected models in parallel: a model concerning the requirements formulated by the legislator in the law documents, a model with a design of the business services that will be implemented by the governmental organization in order to “bring” the law’s business rules to relevant citizens and/or companies, and a model containing the design of the operational business process, and a model containing the implementation of these business processes within the organization. The different stakeholders involved in creating these models can work together in parallel and use each other’s input. Collaboration techniques help them to discuss about the design and to reach consensus about contradicting opinions and concerns. Because each stakeholder have its own “language and abilities to handle abstract models”, the environment emphatically respects the variety of stakeholders. Therefore, stakeholder specific ‘views’ or ‘design studios’ are developed.

The studio for legal experts

One of the views will support the legal expert. In this view the legal expert has the ability to track new or changing legislation. The studio allows the legal expert to break down the legislative text in separate contexts. Each context can be worked out separately, can be re-used in multiple business services and can be related to other

contexts. The studio also enables the detailed, semi-formal description of the semantics (the definition) of the terms used in the texts, and of the annotation fragments

that contain relevant specifications, such as actors, calculation-rules, decision-

The screenshot displays a web-based interface for legal experts. At the top is a navigation bar with tabs: 'Welkom', '1 Analyse wetgevingskader', '2 Analyse intern beleid', '3 Ontwerp diensten en producten', '4 Ontwerp Gestandaardiseerde activiteiten', and '5 Toewijzen en implementeren diensten'. The main content area is divided into several sections. On the left, there's a sidebar with 'Wet inkomstenbelasting 2001' and '3 Memoire van toelichting'. Below this is a circular diagram representing a network of concepts. The central part of the screen shows a detailed view of 'Artikel 3.155. Te conserveren inkomsten buiten beschouwing' and 'Artikel 3.156. Vragen van een beschikking'. It includes the full text of these articles and a list of numbered points. On the right, there's a 'Context' panel with a diagram showing concentric circles labeled 'Wet inkomstenbelasting 2001', 'Inkomsten uit onderneming', and 'Inkomsten uit arbeid'. Below the context panel is a 'Definities' panel with a list of 'Modaliteiten' including 'Rechten', 'Plichten', 'Bevoegdheden', 'Procedures', and 'Verboden'. At the bottom, there's a form titled 'kan een verzoek indienen bij de inspecteur.' with fields for 'Recht indienen verzoek bezwaar', 'Recht', 'Actie Belastingplichtige als ondernemer', and 'Aanvullend'.

rules, legal rights, obligations, procedures and so on. It will also enable visual modeling of the hidden structures that specify how decisions or calculations should be made.

During creation of a law, the proposed law text may change many times. The system will act on automatically sent change alerts. It will intelligently compare the text of newer versions, will visualize changes made and will transfer the unchanged parts of the model from the old version to the new version. Besides comparison, active support is available for determining the impact of changes on the design specification already available, and on operational processes and systems.

The business service design studio for business architects

Besides the view for the legal expert, a view is created for the business architect responsible for determining and designing the organization's business services. This view uses its own sources (documents), but will also use model elements being created by the other stakeholders, like the legal expert. However, these model elements will be presented in the domain specific language of the business architect. For example, the model element "*legal right of a citizen*" will be represented as a "*deontic modality*" to the legal expert whereas it will be represented as a potential "*business event*" toward the business architect, and so on. By translating the model elements in line with one 'mental model' to those in line with that of another type of stakeholder, stakeholders with different backgrounds and languages still can work on one interlinked and consistent, hidden overall model.

In the future, additional views will be added for the other stakeholder types involved, such as IT-analysts, information architects, managers of data administrations, and so on.

Enable collaborative shared decision making

A set of collaborative techniques are combined in an online workspace that is available in and across all "stakeholder views". It will allow modelers as well as more indirectly involved stakeholders to engage in various forms of digitally mediated, dedicated conversation: discuss, react on each other's arguments and opinions, and so on.

The need for games

In our explorative designs and evaluations it became clear that exclusively adopting collaborative modeling techniques to share ideas and information is not sufficient. When designing the business service and its products another important stakeholder comes in sight: the customer, being a citizen or an professional

organization that will be confronted with the services and products based on legislation. Important decisions have to be made such as: “who is our target group exactly?”, “what is the profile of our customer?”, “which criteria should be met in order to entitle individuals within this target area to the products made available by law, like subsidies?”, “which questions should we ask? We cannot always use the terms used in the legislation text, because people might not be able to answer them due to the high level of abstraction.

In order to reach an optimal design, serious games, like for instance a “mystery game” can play an important role. In the mystery game a set of “mysterious” stakeholders (panel members) are trying to ask with a minimal set of questions enough information of the mystery customer in order to find out whether he/she is entitled for getting e.g. a subsidy. Of course each stakeholder use the law and internal policies to formulate the questions. The resulting profile and dialog leads to customer friendly design decisions.

Discussion and future directions.

In our explorative designs and evaluations, the collaboration techniques supporting and structuring such conversation turned out to be a crucial success factor in tackling most issues experienced in the field today. As discussed, they help to gather a solid basis of understanding between all stakeholders. They help to improve efficiency in decision making. They offer the possibility to share knowledge between individuals and they help to implement transparent business processes. Although these positive aspects will not come as a surprise for the community of computer supported collaborative work, they are completely new terrain for the business rule management community however.

The new role of collaboration in governance will have impact on the organization and tools developed to support the collaborative process. First of all a direct reference should be created between discussions and the business rules that are being produced. Also it is necessary to select which discussions may be used for the governance process. E.g. is it desirable to include the names of the stakeholders involved in the formulation of the business rule or should this be anonymous or accessible for special lawyers only? This will be an important issue for further study.

Important additional success factors may be found in further refinement of the basic techniques by means of additional visualization, games and facilitation support.

There is no effective discussion without a clear understanding of the problem. Visualization of developed policy is crucial to evaluate the effectiveness and impact of the business rules formulated.

As discussed, serious games are considered to be an additional means of enhancing the outcome for a problem. Game elements combined with effective

visualization can help stakeholders discover the best operational business policy together. Rules of play can also help guide problems collaborative solving processes and conceptualization. In addition, game elements can help motivate participants and make goals and progress more visible and manageable.

Besides diverging techniques like sharing ideas and opinions there is a strong need for converging techniques in order to make actual decisions one can base further action on. In short, there is need for facilitation. Explorative investigations in collaborative modeling setups, in the case project but also, for example (Hoppenbrouwers & Rouwette, 2012), showed that rules are needed to structure and facilitate the group decision making process. Many of these rules deal with social factors like handling different levels of experience and power positions between stakeholders involved. Facilitation is a skill and this capability is often scarcely available within organizations. Because of the continuous process of translating large scale legislation into specifications, an important next step is to investigate whether it is feasible to add computer aided facilitation techniques to the platform, in order to meet the crucial needed scalability.

References

- de Vreede, G. J., & Briggs, R. O. (2005). Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks. *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS-38)*. Big Island, HI, USA: IEEE Computer Society.
- Hoppenbrouwers, S., & Rouwette, E. (2012). A Dialogue Game for Analysing Group Model Building: Framing Collaborative Modelling and its Facilitation. *International Journal of Organisational Design and Engineering, special issue on Collaborative Modeling*, 2(1), 19-40.
- Hoppenbrouwers, S., Schotten, B., & Lucas, P. J. F. (2010). Towards Games for Knowledge Acquisition and Modeling. *International Journal of Gaming and Computer-Mediated Simulations, Special issue on AI and Games*, 2(4), 48-66.
- Kolfschoten, G., Briggs, R., de Vreede, G. J., Jacobs, P., & Appelman, J. (2006). A conceptual foundation of the thinkLet concept for Collaboration Engineering. *International Journal of Human-Computer Studies*, 64, 611--621.
- Wyner, A., Engers, T. v., & Bahreini, K. (2010). From Policy-Making Statements to First-Order Logic. *Electronic Government and the Information Systems Perspective* (pp. 47-61): Springer.

Using natural user interfaces for collaborative process modelling in virtual environments

Erik Poppe, Jan Recker, Daniel Johnson, Ross Brown

Queensland University of Technology, Australia

e.poppe@student.qut.edu.au, j.recker@qut.edu.au, dm.johnson@qut.edu.au,
r.brown@qut.edu.au

Abstract. Modelling business processes for analysis or redesign usually requires the collaboration of many stakeholders. These stakeholders may be spread across locations or even companies, making co-located collaboration costly and difficult to organize. Modern process modelling technologies support remote collaboration but lack support for visual cues used in co-located collaboration. Previously we presented a prototype 3D virtual world process modelling tool that supports a number of visual cues to facilitate remote collaborative process model creation and validation. However, the added complexity of having to navigate a virtual environment and using an avatar for communication made the tool difficult to use for novice users. We now present an evolved version of the technology that addresses these issues by providing natural user interfaces for non-verbal communication, navigation and model manipulation.

Introduction

Virtual worlds have received continuous attention in industry and research for purposes of training and remote collaboration (Messinger et al., 2009). While there have been studies showing the successful use of virtual worlds for such scenarios, there is also evidence that a key factor in their success is the familiarity of the users with virtual environments (Montoya, Massey, & Lockwood, 2011).

In previous work we have explored the use of virtual worlds for remote collaborative process modelling (Poppe, Brown, Recker, & Johnson, 2013). Process modelling involves the capture and documentation of organizational processes in semi-formal diagrammatic specifications for purposes of execution, automation, analysis or redesign.

In our ongoing experiments on the use of virtual worlds for collaborative modelling, we often note that relatively inexperienced users find using a mouse and keyboard to navigate the

virtual environment difficult, in turn impeding their ability to focus on the task at hand – process model creation, analysis or manipulation.

Prototype

We have previously presented a prototype process modelling tool (Figure 1) that uses a virtual environment and avatars to enable visual cues such as pointing and gesturing to facilitate communication between remote collaborators (Poppe et al., 2013). In this tool, collaborators control representations of themselves in a 3-dimensional space to view, create or manipulate process model elements such as tasks, events, gateways or other.

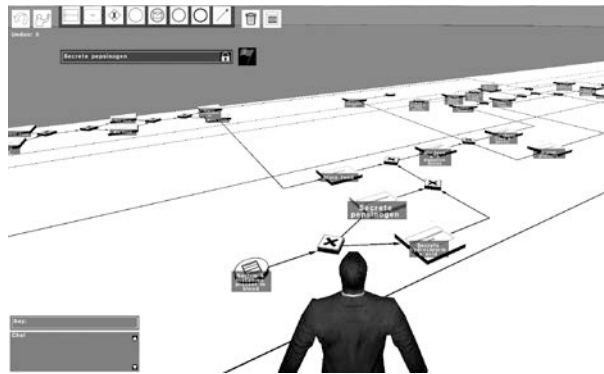


Figure 1. BPM Virtual Modeller (Poppe et al., 2013)

While our evaluation of this tool remains ongoing, we have identified three general usability issues in regards to user interaction with our tool:

- How do we animate an avatar with a large degree of freedom intuitively?
- How do we make model manipulation as intuitive as possible?
- How do we make navigation as intuitive as possible?

Answers to these questions have been suggested in the HCI and Virtual Worlds literature (Marks, Windsor, & Burkhard, 2012; Mazalek et al., 2011; North et al., 2009). On basis of this research, we have implemented *body-tracking*, *head-tracking* and *touch input* to address the issues described¹.

We address the first issue by tracking the posture of the user with a consumer depth camera and applying this posture to the avatar in real-time. This approach enables the use of gestures such as waving, gesticulation, and head nods and shakes, without having to navigate menus or remember buttons. Furthermore, this input method allows for gestures that have not been predefined or subtle variations of typical gestures.

Second, we have implemented a feature for model manipulation via touch input. Instead of having to use a mouse to edit the process model, the user can touch the process model on the screen to create, move, scale or delete elements.

A final issue of using a virtual world compared to other modelling tools is navigation. The 3D view requires users to navigate their virtual bodies in the virtual space, both for

¹ A demonstration video of the prototype can be seen at: <http://www.youtube.com/watch?v=nvfoBfWpxKU>

communication purposes and for viewing all parts of the model. This navigation requires movement along 3-axes and rotation around 2-3 axes and is commonly achieved using a combination of a mouse for rotation and multiple keyboard keys for movement. In our experience, this is confusing for users with limited experience with virtual environments. We therefore implemented head-tracking for camera control. The user can now turn her head to either side, up or down, to have the view turn that way and move his head forward, backwards, sideways, up or down to move the camera in the according direction.

Between these input methods, the keyboard is now required only for labelling model elements. Also, we have reduced the number of keys the user needs to remember to a bare minimum. These interfaces should now enable users to focus on the task of collaborative modelling instead of handling input devices.

Conclusion

We have presented a prototype virtual world that uses natural user interfaces to minimize usability issues for users that are unfamiliar with virtual worlds. In our ongoing work we are executing experimental tests to examine (a) whether this interface indeed facilitates the use of the tool by novice users, and (b) how collaborative modelling processes are enacted by users in a virtual environment.

Acknowledgments

The contributions by Erik Poppe to this research have been supported by the Smart Services CRC in Australia - <http://www.smartservicescrc.com.au>.

References

- Marks, S., Windsor, J., & Burkhard, W. (2012). Head Tracking Based Avatar Control for Virtual Environment Teamwork Training. *Journal of Virtual Reality and Broadcasting*, 9(9).
- Mazalek, A., Chandrasekharan, S., Nitsche, M., Welsh, T., Clifton, P., Quitmeyer, A., Peer, F., et al. (2011). I'm in the Game : Embodied Puppet Interface Improves Avatar Control. *International Conference on Tangible, Embedded, and Embodied Interaction* (pp. 129–136). Funchal, Portugal: ACM.
- Messinger, P. R., Stroulia, E., Lyons, K., Bone, M., Niu, R. H., Smirnov, K., & Perelgut, S. (2009). Virtual worlds — past, present, and future: New directions in social computing. *Decision Support Systems*, 47(3), 204–228.
- Montoya, M. M., Massey, A. P., & Lockwood, N. S. (2011). 3D Collaborative Virtual Environments : Exploring the Link between Collaborative Behaviors and Team Performance. *Decision Sciences*, 42(2), 451–476.
- North, C., Dwyer, T., Lee, B., Fisher, D., Isenberg, P., Robertson, G., & Inkpen, K. (2009). Understanding multi-touch manipulation for surface computing. *Human-Computer Interaction - INTERACT* (Vol. 5727, pp. 236–249).
- Poppe, E., Brown, R., Recker, J., & Johnson, D. (2013). Improving Remote Collaborative Process Modelling using Embodiment in 3D Virtual Environments. *Asia-Pacific Conference on Conceptual Modelling*. Adelaide, Australia.