Separation of Concerns in Model Elicitation – Role-Based Actor-Driven Business Process Modeling

Stefan Oppl[1] and Thomas Rothschädl[2]

[1] Department of Business Information Systems – Communications Engineering
Kepler University of Linz, Linz, Austria
stefan.oppl@jku.at
http://www.ce.jku.at

[2] Metasonic AG, Münchner Straße 29, Hettenshausen, 85276 Pfaffenhofen, Germany
thomas.rothschaedl@metasonic.de
http://www.metasonic.de

Abstract. Elicitation of business process knowledge can be facilitated by visualization of conceptual process models. 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 processes 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 visualization and modeling support that enable a modeling process distributed across role while maintaining one consistent overall model representation. A prototypical implementation of these concepts using distributed tangible tabletop interfaces is presented and results of exploratory tests are discussed. Based on this results the introduced concepts are refined end extended together with an industry partner to create a table top device which can be used in real world model elicitation scenarios.

Keywords: Visualization Techniques for Collaboration and Distributed Processes, Elicitation of Process Knowledge, Tangible Tabletop Interfaces.

1 Introduction

Business Process Models are a recognized means for representation of knowledge about work in organizations [11][18]. They can be used for asynchronous communication of information about business processes [19] and also facilitate elicitation and alignment of business process knowledge [26].

Work is an inherently cooperative phenomenon [30] with activities distributed over different actors. These actors perform their contribution to the overall process in different roles and communicate with each other to pass on their work results [29]. Capturing information about work thus has to involve all relevant

C. Zehbold (Ed.): S-BPM ONE 2014, CCIS 422, pp. 3–20, 2014.

stakeholders to form a sound model of the work process as carried out in organizational reality [31][18]. The process of capturing knowledge about work in a business process model is a form of explicit Articulation Work [30]. It includes the externalization and alignment of different views on work processes from all involved actors [15] and is an collaborative activity itself.

Recent research in the area of collaborative business process modeling (cf. 2) focuses on means of support for collaboration and negotiation in physical or virtual shared spaces. The visualization of the model in general is shared among all participants and presents an overall view on the process. A shared overall view, however, might cause unnecessary cognitive load during elicitation of process knowledge [4]. Allowing actors to focus on their individual role in a process (i.e. their activities and communication with others) in contrast leads to more detailed and refined models that better reflect the actual perception of their work [5]. The objective of this work is to develop model visualizations and elicitation methods that enable capturing process knowledge separately for each involved role while maintaining one single overall model of the process.

The remainder of this paper is structured as follows: the next Section gives an overview about the current state of art in support for collaborative business process modeling. It describes how previous research has approached model visualization in spatially distributed settings and identifies modeling support relevant to the approach examined here. Section 3 elaborates in more detail on the notions of role and actor in the context of collaborative work and modeling processes and discussed requirements on suitable modeling languages. Based upon this conceptualization, different modes of model visualizations are proposed to support modeling of different interaction scenarios in collaborative processes.

Section 4 presents a prototypical implementation of the visualization concepts and describes a showcase to explore user interaction with respect to upon the developed visualizations. The Section closes with a description of the shortcomings that have been identified so far. Based on the identified shortcomings, possible solutions are discussed to enable an operation of the presented system under real world conditions. The paper closes with future directions of research.

2 Current Support for Collaborative Modeling Processes

Collaborative modeling of business processes is a field of research that has gained visibility in the last years. Several systems have been proposed to support collaborative modeling processes in different co-located and distributed settings. Model visualizations that separates a model into distinct parts along the involved roles inherently require a spatially distributed approach to modeling. In order to maintain a sound overall model, the distinct model parts should be created synchronously to allow for immediate interlinking and alignment of model parts. We therefore review the current state-of-the-art in cooperative business process modeling with a focus on solutions that target spatially distributed settings for synchronous modeling.

CEPE [28] was one of the first real-time cooperative modeling applications available and has been designed to support business process reengineering use-cases in distributed settings. All users share the same model visualization which is propagated synchronously to the attached modeling software.

Decker and Weske [6] present a tool to cooperatively manipulate BPMN models on a web-based platform, on which all participants share a common view on the process. SAP has presented similar functionality in their Gravity-system that is based upon Google Wave technology for synchronously propagating model changes [8]. Hahn et al. [14] have examined the effects of the same system on collaborative process modeling in distributed settings in an exploratory study and identified current shortcomings of the prototype. Those were mainly related to lacking means of communication and shared access to common information during the modeling process. Participants also requested clear guidelines on how to structure the process and use BPMN elements to model collaborative behavior. The shared modeling surface caused conflicts in concurrent modeling, which could not be resolved due to lacking modeling space.

Brown et al. [2] present a modeling approach for BPMN using virtual 3D-worlds. Collaboration support is not directly anchored on the model but shifted to the surrounding virtual environment that facilitates immediate interaction and communication even in distributed settings.

Dollmann et al. [7] have focused on transforming models on the fly to different semantically enriched representations, also including a transformation of the graphical notation. They present a procedural model to collaboratively develop cross-domain process models with a focus on semantic mapping and do not focus on collaboration support in their first prototype.

Riemer et al. [25] have examined a set of 12 commercially available business process modeling tools regarding their support for collaborative modeling. While they commonly found support for asynchronous modeling and concurrent modeling of independent models, none of the examined tools supported synchronous modeling of business processes in distributed settings.

Reviewing the current state-of-the-art, collaborative modeling of business processes so far has mainly been addressed in settings, where a model is manipulated concurrently by several users in spatially distributed editors. Approaches that explicitly support temporally asynchronous [9] or spatially co-located settings [24][16] have been omitted here, as their applicability for the use-case described here is limited. Although this work a different approach to visualization than all other approaches to collaborative process modeling, the requirements on support for the modeling process in general sustain. This work therefore draws from prior research mainly through adopting the following requirements:

- Provide means for communication about the modeling process, desirably not only via text but also using audio or video channels [14] or even non-verbal signals [2]. Directly anchoring communication on model elements allows for easier referencing of the points of discourse [6].
- Provide access to all information relevant for modeling all of the time [14]. In a setting, where model parts are created spatially distributed and no overall

view is available for all participants by default, this implies that the actors have to be provided at least with all model information that directly affects them (e.g. the behavior of other roles they are interacting with).

3 Role-Based Process Model Elicitation

Designing means of support for role-based elicitation of collaborative processes requires a detailed understanding of the entities involved in collaboration and their contributions to the overall process. In the first section of this paper, the phenomenon of collaborative work was described [30][29] and the notions of "actor", "role", "activity" and "communication" were introduced in this context. These notions are revisited here to more exactly specify the relevant concepts in the context of this work.

Actors are considered to be individuals active in an organization. Activities are carried out by an actor without immediate interaction with others. Activities of different actors happen in parallel and are coupled with each other 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.

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:

- individual activities carried out by an actor (including decisions)
- communication acts to link individual activities of different actors
  - outgoing communication acts, i.e. actively sending work results
  - incoming communication acts, i.e. receiving work results

In general, (collaborative) business processes are not tailored towards one specific actor but are specified to be carried out by a set of interacting roles. A role is an area of responsibility in the business process at hand. Consequently, several actors are able to take a certain role in a business process. A role, per definition, can only be taken by one specific actor in a specific business process instance (i.e. there are no roles that involve several actors simultaneously). This does not prevent actors to be basically able to take different roles. Communication acts are carried out among roles and interlink the activities carried out by actors acting in a certain role.

Introducing roles in a business process as an abstraction from actual actors introduces another distinction relevant for supporting the elicitation process:

- roles that are represented by only one actor during elicitation
- roles that are represented by several actors during elicitation

Before elaborating on possibilities for visualization model information suitable for role-distributed elicitation, suitable modeling languages have to be identified. In order to allow for visualizing a model distributed along the roles involved in the process, the used modeling language has to provide modeling constructs that allow for structuring the model along these boundaries [1].

3.1 Suitable Modeling Languages

Languages for representation of business processes in general follow different approaches along which conceptual dimensions information is described. A modeling language, that provides constructs to use the "who"-Dimension [34] as the primary factor of structuring, enables to separate areas of concern in a model of a cooperative business process [12]. An overview about suitable modeling languages (without intending to be exhaustive) is given in the following:

Role-Activity-Diagrams (RAD) [23] are an early approach to structure business processes along roles in a business process. They provide "roles" as constructs for structuring activities along areas of responsibility and "interactions" to model communication among roles. Interactions, however, are always considered to be acts of synchronization and thus do not allow for sending messages asynchronously.

In UML Activity Diagrams [3], "partitions" can be used to distinguish roles (although they are semantically not restricted to represent roles). "Flows" are used to connect activities. The is no separate semantic construct to distinguish among flows within a partition and among partitions.

BPMN [33] provides "pools" and "lanes" to structure processes along areas of responsibility. "Message Flows" are a construct to explicitly model communication among pools - they cannot be used among lanes. For the use-case proposed here, mapping roles to BPMN-pools would be an appropriate decision. Message flows originate in sending message events and end in receiving message events. The necessary model elements specified above thus can be fully mapped to BPMN.

S-BPM [10] follows an approach very much in line with the concepts proposed above. Models consist of "subjects" that interact using "messages". Subjects basically maps to the concept of roles described above. Their behavior is modeled using "action states", "sending states" and "receiving states", where the latter two are used to send and receive messages respectively. S-BPM element thus also fully cover the necessary modeling elements described above.

Summarizing, BPMN and S-BPM are both suitable to implement the model visualization approach described in this paper. The concepts for visualization presented in the next section are language-independent and can be implemented using either BPMN or S-BPM (or any other language fulfilling the fundamental requirements).

3.2 Concepts for Role-Based Model Visualization and Modeling Support

Separating a process along the involved roles requires a number of support measures relevant for interlinking and aligning different views on a business process and ultimately deriving a commonly agreed upon model of the business process.

The role-based areas of concern are interconnected by communication processes. As described in Section 3.1, communication processes are generally represented by flows of discrete messages that are sent from one role to another to trigger certain behavior at the receiving end.

Modeling of Role Behavior. Each role's contribution to work is visualized 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 should be implemented [15]. These different viewpoints require alignment in order to derive one single, commonly agreed upon view on a business process. Consequently, collaboration support for modeling role behavior has be provided. All participating actors in this case share the same view on the role's part of the model. Shared views during collaborative process modeling have already been addressed in literature (cf. Section 2). Solutions for both, distributed (such as as [8]) or co-located settings (such as [16]) are viable here.

Following the argumentation at the beginning of this section, modeling elements for activities, decisions and communication acts are required. All modeling languages mentioned above provide the required set of elements (e.g. tasks, gateways as well as sending and receiving message events in the case of BPMN). Elements to model communication require special treatment, as they conceptually span across roles and require visualization for both involved roles (cf. Figure 1).

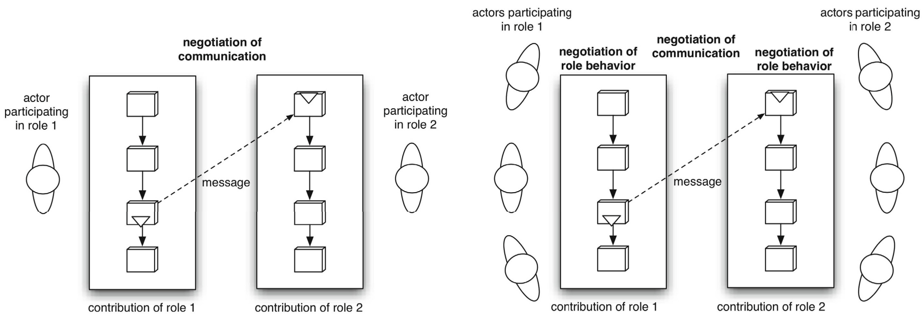

Fig. 1. Distributed manipulation of role-separated model with one actor per role (left) or multiple actors per role (right)

Modeling of Communication Acts. Communication among roles occurs whenever results of work (data and/or physical goods) have to be passed on from one role to another. In the following, the notion of "message" is used for these results of work. From a modeling process perspective, the following situations can occur:

– send a message to another role
– get notified that a message has been sent to one's own role
– request a message from another role to be able to proceed with one's own part of the process
– get notified that another role requests a message to be able to proceed with its part of the process

The first two situations occur regularly during the modeling process and would be sufficient to describe all communication situations if the business process was modeled in fully sequential manner across all involved roles. This would require the actors modeling a certain role to wait for another role to finish its work and send its result, before they can proceed modeling with their own process, if they are dependent on these result.

The third and fourth communication acts have been introduced to avoid those delays in modeling. Actors can specify messages they expect to arrive from another role and continue modeling as if this message already would have arrived. Aside from decoupling the roles' modeling processes the possibility to request messages also allows to uncover unclear communication flows or inconsistent expectations of who has to communicate which information to whom under which circumstances.

Whenever a message is created, either by sending a message to another role or requesting a message from another role (i.e. creating a local message proxy), its respective counterpart message has to show up at the communication partner's model side. Incoming messages or message requests however do not necessarily need to be processed immediately. For that reason, they are pooled in message trays that visualize all unprocessed messages separated by the according communication partners (for an example, see Figure 2).

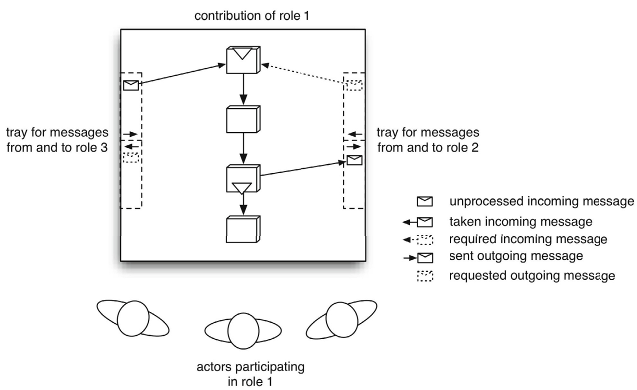

Fig. 2. Conceptual model visualization elements for one role

Communication about the Modeling Process. During the review of prior work in the area of collaborative business process modeling (cf. Section 2), providing means for communication about the modeling process has been identified as an important requirement for modeling support. Approaches to facilitate communication in all its aspects in distributed settings have been extensively reviewed in related work and are beyond of the scope of this paper. An important aspect to be considered when selecting means of communication support for the specific setting described in this paper is actors taking different roles by default do not see the same information and thus might have higher demand for communication about the modeling process and require more powerful tools for communication. This issue can be partially overcome with means to promote distributed model awareness as described in the next paragraph.

Distributed Model Awareness. The second requirement identified in Section 2 was to ensure availability of information about the process to be modeled and the current state of the model itself. Aside from measures like shared document repositories etc., the requirement needs to be considered more closely for the distributed model visualization used in this work. A view on the business process spanning across role boundaries is useful to develop an understanding about the overall structure of the process. Modeling is limited to the boundaries of a role in the approach presented here. While model visualization by default only contains a role's behavior and its direct communication with other roles, the model representation contains an overall model, thus allowing to visualize other model aspects.

However, which information is necessary about parts of processes affecting other roles, depends on the process to be modeled, i.e. is dependent on its complexity and potential restrictions due to confidentiality (especially relevant in cross-organizational settings, which are not explicitly excluded here). Following the modeling elements specified above and the partitioning approach, the following views can be identified (and be combined freely depending on the use-case):

- view on the overall communication acts (i.e. who is communication with whom about what), potentially including message content
- view on all role's behaviors (i.e. the detailed models of a role's contributions to the process), including all communication acts (i.e. flattened model of the overall process)
- view on the behavior of a role's direct communication partners (e.g. to follow up, how one' sent message are processed or received messages are being created)

4 Implementation of First Prototype

In a first attempt to implement the requirements described above, we have set up a role-based modeling environment based upon an existing interactive tabletop modeling system (cf. [21] for further details). The original system allows for synchronous co-located collaborative modeling and thus fits well the requirement to allow actors to collaboratively specify a role's behavior. The use of a physical tabletop setting for business process modeling is also in line with the positive experiences of Grosskopf et al. [13] in their pilot-study using non-electronically augmented tabletop business process modeling.

We here report on the system setup that has been developed to implement the distributed modeling environment and map its visualization approaches to the concepts described above. Additionally, we present our findings from a first round of exploratory tests that have been conducted to evaluate the applicability of the toolset for distributed collaborative modeling processes.

4.1 System Setup

The tabletop system has been used in a prototypical setting for modeling of business processes distributed across the involved roles. The result of the modeling process is a single process model representation containing the roles' contributions to the process interlinked via their communication acts. During the modeling process, however, the behavior of each involved role and its interaction with others has been modeled separately. A separate interactive table is used for the modeling process of each role.

In its recent iteration, the original system has been extended to support distributed modeling processes [32]. Multiple tables are used for synchronous modeling in a spatially distributed setting (cf. Figure 3). They are technically coupled using a message-based communication infrastructure (for details cf. [32]).

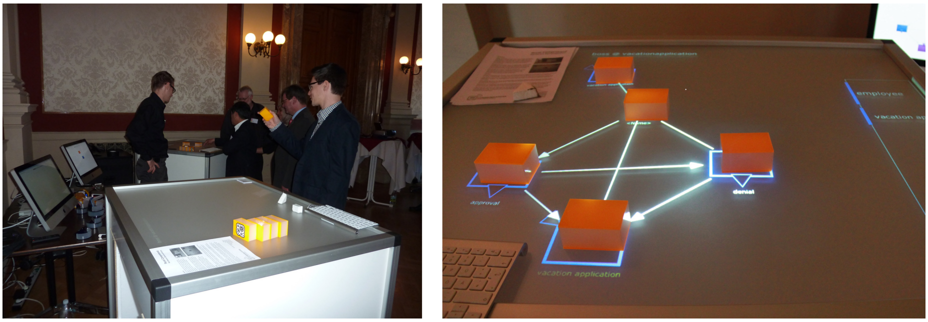

Fig. 3. Tabletop modeling system: two table setup with one table in the front and one in the back (left), model and communication tray of one table (right)

The table allows spatially distributed modeling of different parts of a business process model, which can be coupled via messages [20]. As for BPMN, each table represents one pool with its own sequence flow, while the interaction among pools is modeled by sending messages to represent the message flow. As for S-BPM, the tables map to subjects with the communication behavior being again represented by sent and received messages. In the following paragraphs, the mapping of concepts for role-based modeling support to actual features of the system is presented.

Modeling of role behavior is realized using one generic modeling element, that is used for representing activities, sending and receiving acts. Its semantics and the according visualization changes according to the use of the element in the model. Visualization is altered if an element is used to send or receive message (examples are visible on the right image in Figure 3) The advantages of using a generic modeling element during elicitation is discussed in [32].

Modeling of communication acts uses the communication trays proposed above. All tables are bidirectional interlinked using trays, that display unprocessed incoming messages and provide an area to send messages, that also contains visualizations of not yet processed requested messages. If a generic modeling element is placed in the sending area of the tray, its visualization and semantics are changed to become a sending message element and the name of the message can be specified. If placed next to a requested message element, the according message is sent. The same process is used to process incoming messages. A separate area in the incoming message tray allows to request messages from other roles. In this case, a message proxy is created and can be used for modeling. Modeling elements that are used for interaction with the trays do not affect the role behavior visualization, even if they already have been linked with other elements. The element temporarily removed from its position is replaced by a proxy visualization and allow for exact reconstruction of the actual model state.

Distributed Model Awareness is provided by different model visualization provided on the information display screen. By default, the messages exchanged with the immediate communication partners are graphically visualized. Actors can switch to a global communication view that visualizes all exchanged messages.

4.2 Exploratory Testing and Identified Shortcomings of the Prototype

Based upon findings from a first exploratory study [32] with a first prototype of the system, a second round of tests has been conducted in the course of a conference on subject-oriented business process management. Accordingly, the modeling approach used in the examination has been S-BPM. The aim of the test was to test the comprehensibility of the model visualization and identify areas for further improvement in terms of modeling support and model awareness.

The tables had been deployed in a co-located setting (cf. Figure 2, left). A scenario involving three roles had been prepared, where the behavior of one role was pre-scripted and simulated by a software component. The remaining two roles were assigned to one table each. The basic flow of activities as well as the necessary communication among the roles was provided textually separately for each role during the modeling sessions. Each role contained two to four activities, at least one act of sending and receiving messages and one decision. Prescribing the process to be modeled prevents examination of the systems effects on externalization and negotiation of meaning. A given modeling scenario, however, allows to focus on identifying issues in understanding the elements of the model visualization, their use and awareness about the model parts created on distant tables.

The system was deployed over a period of two days in an openly accessible area at the conference location. In total 42 people in 10 groups of 2 to 6 persons participated in the tests. All participants at least had fundamental knowledge in S-BPM. None of them had worked with the tabletop system earlier. They were briefly introduced to the system's features and modes of interaction and were asked to model the scenario afterwards. Data was collected via observation of the interaction with the system by a supervisor and qualitative feedback by the users after the modeling process.

Due to the co-located setting, support for communication about the modeling process has not been used. Additionally, message content negotiation has not been part of the modeling scenario and also was not used. The feature to request messages had not yet been implement at the time of the development, thus messages could only be used synchronously.

Initial Findings. The usefulness of the system for externalization and collaborative alignment of process knowledge in conceptual models has already been shown for co-located settings in earlier work [22]. The current prototype basically also met the expectations in terms of usability in the conceptually and spatially distributed setting presented here. In our tests, all groups except one were able to create a role-distributed model of the provided scenario after a brief introduction and occasional guidance in case of technical problems. There have been no observable or communicated misunderstandings in how to create models of role behavior or to send and receive messages. Some shortcomings of the current toolset, however, have been identified:

- **Asynchronous Request of Messages:** The need to wait for incoming messages in order to continue modeling has been experienced as a major obstacle in the modeling process.
- **View Overall Process:** An overall view of the complete process was hard to maintain but would have been helpful
- **Optimize Space Usage:** More complex models cannot be visualized at all due to space restrictions

5 Towards Deployment in Daily Business

The previous section identified shortcomings of the prototypical implementation based on an exploratory user study. This section presents possible solutions to create an enhanced system which can be examined under real-world settings together with the industry partner *Metasonic AG*. Therefore, practical issues such as logistics, maintainability and operability by business users had to be considered.

5.1 Adaptions for Business Use

Besides the conceptual issues, examined in Section 4.2, also some adaptions are necessary, to engage the industry partner in the research project.

The presented prototype needs to operate under controlled lightning conditions. In case of changing lightning conditions a correct detection of the graspable modeling elements could suffer. One additional issue was the assemble and disassemble of the tabletop interface itself. This was time-consuming, error prone and demands at least basic skills in craftsmanship. The solution to the mentioned hardware issues was to enable the software to operate also on professional hardware base from third party supplier which also support the TUIO protocol [17] for processing user interactions on the table top interface. Due to this, future research could consider user interaction concepts separately from hardware issues.

To enable the integration between the tabletop interface and the BPM-Suite from the industry partner, the communication server [32], which uses the Extensible Messaging and Presence Protocol (XMPP) [27], is replaced by the centralised model repository from the industry partner. As the centralised model repository stores an overall model representation and supports a concurrent modification of the same process by different users, concurrent process modeling on co-located and distributed table top interfaces is still supported. Additional the process modeling environment *Metasonic Build* can be used to visualise the evolution of role behaviour and acts of communications between roles. As the model repository stores the overall process it is possible to show and manipulate the behaviour of all roles on dislocated tables or with the modeling environment. Figure 4 shows different ways of model creation with a screen based modeling tool on a client pc, a co-located modelling session with two tables for elicitation of two roles and a further big screen to visualise communication acts and an other, dislocated table for elicitation of an additional role.

5.2 Resolution of Shortcomings of the Prototype

Additional to the practical changes, conceptional improvements to solve the issues in Section 4.2 were developed in workshops together with experts from the industry partner. The aim of the workshops was to improve the existing system and to be used for process model elicitation under real world conditions.

Asynchronous Request of Messages. As defined in Section 3.2 and as one of the findings in Section 4.2 the system has to be able to request messages in an asynchronous way without the need to wait until the message is created by the sending subject.This can be realised as user interaction pattern by putting a send or receive brick to the corresponding area on the tray. When the element is detected, a textbox opens, which allows to enter a name for the new message. After confirming the name with return, the new message is created and attached to the element for further use within the modeling process. Immediately after creation of the new message, the corresponding subject get aware of it as the new message is added to the tray representing the subject which created the message. Figure 5 shows a screenshot of the table representing *Subject 1*. The message tray on the left side represents the messages *Subject 1* exchanges with *Subject 2*. Additional the information that *Subject 1* sends *Message 1* to *Subject 2* and receives *Message 2* from *Subject 2* is visible for the user.

View Overall Process. The focus of the first prototype was to enable a distributed system where the different roles only exchange messages among each

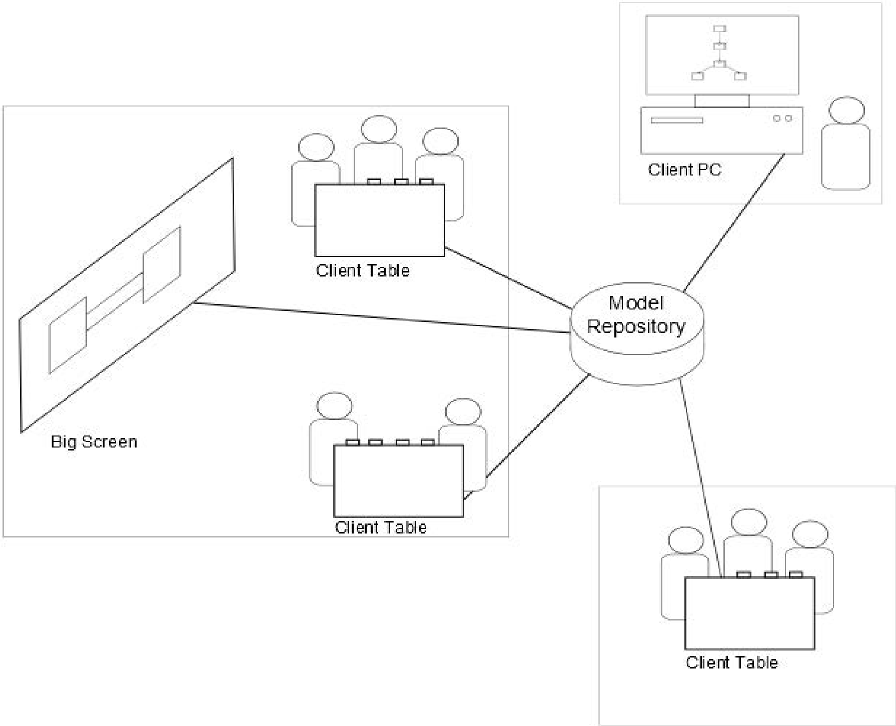

Fig. 4. Overview of possible distributions with a centralised model repository

other. Each modeling table therefore knew only its own internal behaviour. The interaction with other tables was managed and mapped via a centralised chat server where each process elicitation session was represented by a separated chat-room. Each role within the business process was represented by a user on the chat server and the messages, which are exchanged between the roles, have been mapped as chat messages between the users. This fits well to the Separation of Concerns approach, however it lacks in reality as it is not always clear if a certain task has been already performed by a different role during elicitation. Moreover, the experiment in Section 4.2 shows that users were missing an overall view of the process either to be able to identify all existing roles, or to have an overview which interactions happen between these or even to examine the behaviour of one special role. Therefore, it is necessary to be able to visualise, and eventually refine, the already modeled behaviour of other roles of the same process. This can be realised on the modeling table, with the need of removing all tangible elements to load an overview of the different roles and their interaction or to display the behaviour of one specific role. One other possibility would be to access the centralised model repository with an additional device to display the desired view on a extra screen (shown in Figure 4), a projector or even a tablet.

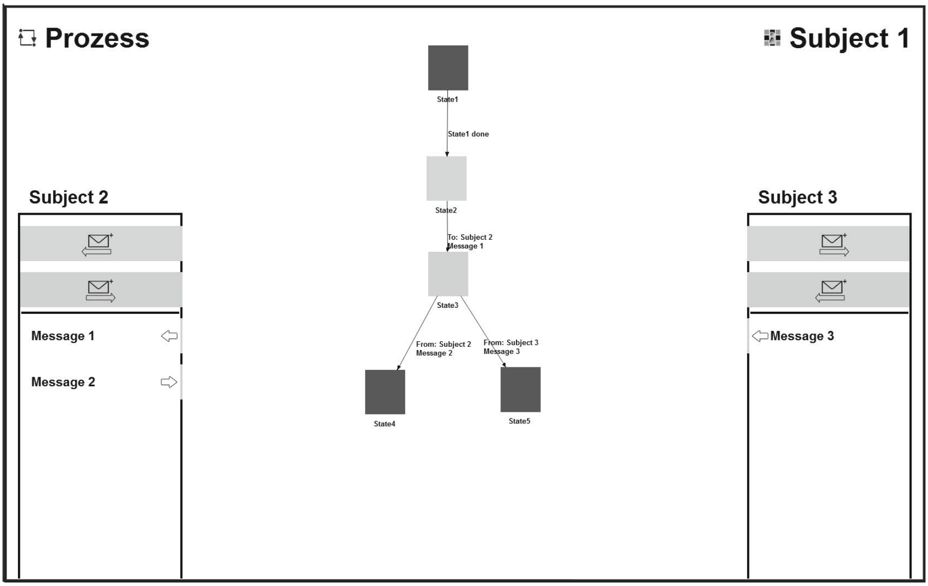

Fig. 5. Screenshot of the table surface with trays for each actor. By putting an element to the corresponding position on the top of each tray, new messages can be created. In the separated area below, already created messages are listed.

Optimize Space Usage. One lack of the existing system is that more complex models cannot be visualized at all due to space restrictions. This problem could be split into two dimensions, which have to be examined in two different ways.

One dimension of complexity covers the amount of different roles and the mapping of their interaction behaviour. Users which uses the examined prototype could only express acts of communication in maximum with two other subjects concurrently. This is due to the fact that the presented approach needs an own interaction tray for each role. Figure 5 shows that *Subject 1* interacts with the two roles *Subject 2* and *Subject 3*. If *Subject 1* also needs to interact with an additional role, *Subject 4* for instance, it has to disable one of the two currently displayed trays (exemplary *Subject 3*) to show a own tray for the additional role *Subject 4*. This has the negative side effect that if the prior disabled role *Subject 3* wants to interact with the role modeled on the tangible interface (*Subject 1*) will only get aware of this if it enables the tray for *Subject 3* again. With the result of having the same awareness problem with *Subject 4*.

A solution for this problem is to change the mapping of the trays itself from separated trays for each role to an own tray for incoming messages and one tray for outgoing messages. The corresponding role has to be shown next to the name of the message. With this approach only two different trays are sufficient to map the message interaction of one role. One for messages sent to the current modeled role and one for messages the current modeled role sends to other roles. Figure 6 illustrates the modeling surface of one table with only two trays. In difference to the Situation shown in Figure 5, the users also get aware that *Subject 4* needs an interaction. It sends *Message 5* to *Subject 1* and expects *Message 6* from *Subject 1*.

For sending a new message the user places a sending brick on the tray area for creating a new message. When the system detects a sending brick on the create message area, a selection menu with all available roles appears next to the brick. By rotating the brick the user has to choose the role which should receive the new message. Then the user has to name the new message. After the message is created it will appear in the tray for incoming messages of the receiving role, which can use it in the own internal behaviour.

It also can happen that a role did not know who is responsible for the message. Therefore it should also be possible that the creator of the new message could select *Unknown* as the receiver of the new message. If this happens, the new message appears in all input trays on the table until one role feels responsible for the new message and uses it in its own process behaviour.

The creation of a new incoming message can be modeled vice versa.

The second dimension covers the limited modeling space in terms of available spacial possibilities for placing elements on the table surface.Regarding this, first solutions such as hiding model complexity by embedding parts of the model in more generic elements or temporal removement of elements which are not relevant for the current elicitation act, have been discussed in [32]. One part of this problem was caused by recognition problems, which have been already mentioned in Section 5.1. The first tables forces users to place modeling bricks in a limited area near the center of the modeling surface to gain a stable marker recognition. Therefore, it was not possible to vacate the table center by putting currently non-essential bricks to the side (but still on the table) without loosing there already elicited, inherent information. As the new hardware base offers accurate element recognition over the whole modeling surface. This fact, in addition with slightly smaller modeling bricks has mitigated this problem.

6 Summary and Future Research

In this paper, we have presented an approach to role-based visualization of business process modeling that aids elicitation of work knowledge from actors involved in the process and facilitates alignment of their individual views on their work. We have derived concepts for model visualization and cooperative modeling support from both, earlier research on collaborative business process modeling and existing modeling languages that allow structuring the process along role boundaries. In a first attempt to implement the visualization concepts, we have created a distributed tangible tabletop interface. Initial results of the testing have been very promising, however, shortcomings have been identified in the examined implementation. Based on this, solutions and improvements to overcome the tracked shortcomings, that have been developed together with an industry partner, were presented. Future work will focus on the iterative refinement and extension of the system. Based upon such a fully usable version of the interactive

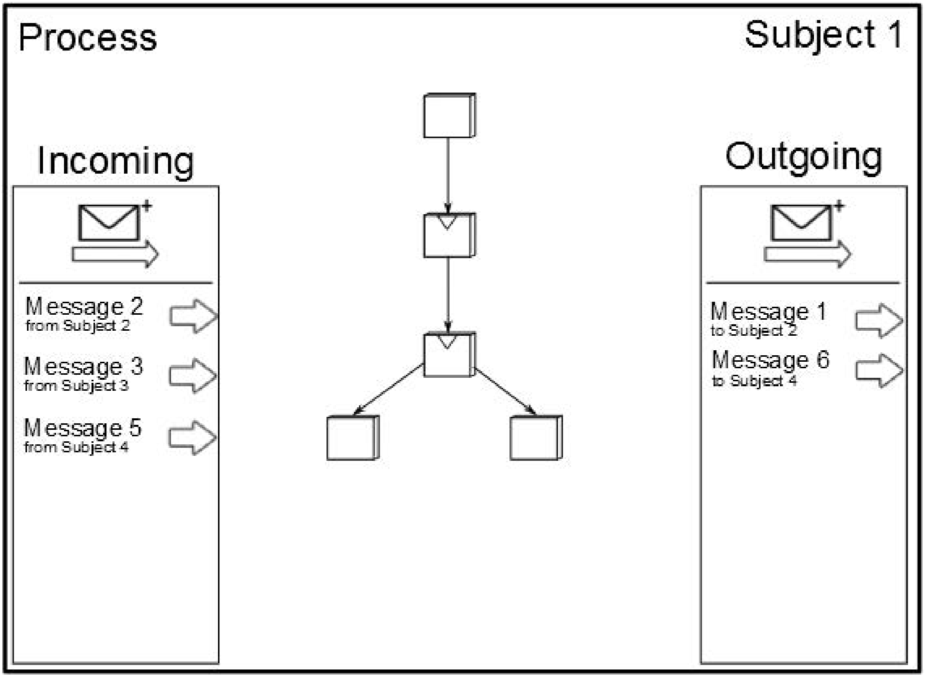

Fig. 6. Surface with separated trays for incoming and outgoing messages. On the top of each tray new messages can be created. In the separated area below, already existing messages are listed.

system, its usefulness in terms on supporting the distributed model elicitation and user interaction concepts can be evaluated. Ultimately, the suitability of the overall concept for process knowledge elicitation will be validated empirically in several case studies that will be conducted in the course of the academic-industry partnership IANES (http://www.ianes.eu).

Acknowledgements. 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^o 286083 (IANES).

References

1. Adamides, E., Karacapilidis, N.: A knowledge centred framework for collaborative business process modelling. Business Process Management Journal 12(5), 557–575 (2006)
2. Brown, R., Recker, J., West, S.: Using virtual worlds for collaborative business process modeling. Business Process Management Journal 17(3), 546–564 (2011)

3. de Cesare, S., Serrano, A.: Collaborative modeling using uml and business process simulation. In: Proceedings of HICSS 2006. IEEE Press (2006)
4. Crapo, A., Waisel, L., Wallace, W., Willemain, T.: Visualization and the process of modeling: a cognitive-theoretic view. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 218–226. ACM Press, New York (2000)
5. Dann, H.D.: Variation von Lege-Strukturen zur Wissensrepräsentation. In: Scheele, B. (ed.) Struktur-Lege-Verfahren als Dialog-Konsens-Methodik., Arbeiten zur sozialwissenschaftlichen Psychologie, vol. 25, pp. 2–41. Aschendorff (1992)
6. Decker, G., Weske, M.: Towards collaborative business process modeling. Cutter IT Journal (2009)
7. Dollmann, T., Houy, C., Fettke, P., Loos, P.: Collaborative business process modeling with comomod-a toolkit for model integration in distributed cooperation environments. In: Proceedings of the WETICE 2011, pp. 217–222. IEEE Press (2011)
8. Dreiling, A.: Gravity – collaborative business process modelling within google wave (2009), `http://scn.sap.com/people/alexander.dreiling/blog/2009/09/02/gravity-collaborative-business-process-modelling-within-google-wave`
9. Erol, S., Mödritscher, F., Neumann, G.: A meta-design approach for collaborative process modeling. In: Proceedings of the 2nd International Workshop on Open Design Spaces (ODS 2010), pp. 46–62 (2010)
10. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subjektorientiertes Prozessmanagement. Hanser Verlag (2011)
11. Gasson, S.: The dynamics of sensemaking, knowledge, and expertise in collaborative, boundary-spanning design. Journal of Computer-Mediated Communication 10(4) (2005)
12. Giaglis, G.: A taxonomy of business process modeling and information systems modeling techniques. International Journal of Flexible Manufacturing Systems 13(2), 209–228 (2001)
13. Grosskopf, A., Edelman, J., Weske, M.: Tangible business process modeling – methodology and experiment design. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 489–500. Springer, Heidelberg (2010)
14. Hahn, C., Recker, J., Mendling, J.: An exploratory study of IT-enabled collaborative process modeling. In: Muehlen, M.z., Su, J. (eds.) BPM 2010 Workshops. LNBIP, vol. 66, pp. 61–72. Springer, Heidelberg (2011)
15. Herrmann, T., Hoffmann, M., Kunau, G., Loser, K.: 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 (2002)
16. Herrmann, T., Nolte, A.: The integration of collaborative process modeling and electronic brainstorming in co-located meetings. In: Kolfschoten, G., Herrmann, T., Lukosch, S. (eds.) CRIWG 2010. LNCS, vol. 6257, pp. 145–160. Springer, Heidelberg (2010), `http://www.sociotech-lit.de/HeNo10-IoC.pdf`
17. Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E.: TUIO: a protocol for table-top tangible user interfaces. In: Proc. of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (2005), `http://mtp.athlabs.com/export/90/docs/TUIO`
18. Lewis, M., Young, B., Mathiassen, L., Rai, A., Welke, R.: Business process innovation based on stakeholder perceptions. Information, Knowledge, Systems Management 6(1), 7–27 (2007)
19. Niehaves, B., Plattfaut, R.: Collaborative business process management: status quo and quo vadis. Business Process Management Journal 17(3), 384–402 (2011)

20. Oppl, S.: Subject-Oriented Elicitation of Distributed Business Process Knowledge. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 16–33. Springer, Heidelberg (2011)
21. Oppl, S., Stary, C.: Tabletop concept mapping. In: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction (TEI 2009). ACM Press (February 2009)
22. Oppl, S., Stary, C.: Effects of a Tabletop Interface on the Co-Construction of Concept Maps. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011, Part III. LNCS, vol. 6948, pp. 443–460. Springer, Heidelberg (2011)
23. Ould, M.A.: Business Processes: Modelling and analysis for re-engineering and improvement. John Wiley and Sons (1995)
24. Renger, M., Kolfschoten, G., de Vreede, G.J.: Using interactive whiteboard technology to support collaborative modeling. In: Briggs, R.O., Antunes, P., de Vreede, G.-J., Read, A.S. (eds.) CRIWG 2008. LNCS, vol. 5411, pp. 356–363. Springer, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-92831-7_29
25. Riemer, K., Holler, J., Indulska, M.: Collaborative process modelling-tool analysis and design implications. In: Proceedings of ECIS 2011, Association for Information Systems (2011)
26. Rittgen, P.: Negotiating models. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 561–573. Springer, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-92831-7_29
27. Saint-Andre, P., et al.: Extensible messaging and presence protocol (XMPP): Core. RFC 3920, W3C (2004)
28. Santoro, F., Borges, M., Pino, J.: Cepe: cooperative editor for processes elicitation. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 10 p. IEEE (2000)
29. Schmidt, K., Simone, C.: Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. Computer Supported Cooperative Work 5(2/3), 155–200 (1996)
30. Strauss, A.: Continual Permutations of Action. Aldine de Gruyter, New York (1993)
31. Stuit, M., Wortmann, H., Szirbik, N., Roodenburg, J.: 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 (2011)
32. Wachholder, D., Oppl, S.: Stakeholder-Driven Collaborative Modeling of Subject-Oriented Business Processes. In: Stary, C. (ed.) S-BPM ONE 2012. LNBIP, vol. 104, pp. 145–162. Springer, Heidelberg (2012)
33. White, S.: Introduction to BPMN. BPTrends (2004)
34. Zachman, J.: A framework for information systems architecture. IBM Systems Journal 26(3), 276–292 (1987)