In: Nolte, A.; Prilla, M.; Lukosch, S.; Kolfschoten, G. and Herrmann, T.: Proceedings of the 1st International Workshop on Collaborative Usage and Development of Models and Visualizations at the ECSCW 2011 (CollabViz 2011)

# Introduction: 1st International Workshop on Collaborative usage and development of models and visualizations (CollabViz 2011)

Alexander Nolte[1], Michael Prilla[1], Stephan Lukosch[2], Gwendolyn Kolfschoten[2], Thomas Herrmann[1]

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

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

[2]Faculty of Technology, Policy and Management, Systems Engineering Department, TU Delft, Netherlands

*g.l.kolfschoten@tudelft.nl*, *s.g.lukosch@tudelft.nl*

## Why Collaborative usage and development of models and visualizations?

The 2011 International Workshop on Collaborative usage and development of models and visualizations is being held as part of the European Conference on Computer Supported Cooperative Work (ECSCW 2011) in Aarhus, Denmark. It brings together researchers investigating the role of models and other visualizations in modern organizations. Given that knowledge on processes and networks as well as flexibility and adaptability for acting in processes and networks becomes one of the most important assets of our economy, the work done by the workshops participants is one of many to follow steps towards understanding and systematically supporting the usage of graphical representations. The importance of research in this area will most likely increase and the content in this volume provides meaningful insights and points to start additional research from.

# Scope and Aim of CollabViz 2011

The usage of graphical representations of static parts of an organization (e.g. diagrams depicting hierarchies in the organization structure or a company's competences) and 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 graphical representations include process models, conceptual models and mind maps, and are used to support multiple tasks such as software development, design and engineering, process optimization and reengineering as well as marketing and strategic development. Obviously, these models are not artifacts used by single users, who develop and use them for their own personal needs. These graphical representations are rather developed for larger target groups throughout an organization to support them in sense making and creating shared understanding. Consequently, they are both used by many people and developed collaboratively, thus being part of and influencing the work of multiple stakeholders in an organization.

Alongside the increasing usage and popularity of graphical representations, there is growing interest in the usage and development of models in the CSCW community. This not only comprises the usage and development by modeling experts, but explicitly takes non-expert users into account. The emerging importance of this new field of CSCW research is reflected by tracks at international conferences (e.g. "Collaborative Modeling" at HICSS 2009, 2010 and 2011), papers at different CSCW related conferences (e.g. Baacke et al. 2009, Brosch et al. 2009, Herrmann and Nolte 2010, Klebl et al. 2009, Prilla and Nolte 2010) and journal contributions (Rittgen 2010, Renger et. al. 2009, Heer et al. 2010, Yuille and Macdonald 2010). Additionally, there are various parallel approaches in familiar research communities such as Group Decision Support, Business Process Management and Group Support Systems.

However, despite the fact that as modeling is a popular approach in practice and thus, many models exist in organizations, they are hardly used by non-experts. Even if they are created collaboratively by process stakeholders they have little impact on the people that are actually working in these processes (cf. Prilla 2010). The reasons for this are twofold. First, there are few insights on the spreading and sustainment of process documentation usage in organizations. Second, up to now little is known about the interaction of non-expert users with models. By interaction, however, we not only refer to the creation of models, but also their usage in people's daily work for purposes such as discussions, knowledge explication and creating a common understanding. This raises questions such as why there is so little use of models after their creation, how this usage can be increased and which kind of tools and modes of interaction are suitable for people who are not modeling professionals.

Besides the usage of models by non-experts, there is an additional research gap in the collaborative modeling of graphical representations. Usually, the collaborative creation of models by non-experts is restricted to collocated workshops and similar modes of interaction and collaboration, where experts facilitate the work and translate non-expert articulations into model or diagram language. 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 and therefore available expertise, these workshops also do not consider the need to support dislocated collaborative modeling. Therefore, we need to find ways to enable ordinary and also dislocated users to contribute actively to the creation and maintenance of models. This may include enabling users to use modeling languages and contribute directly to a model as well as finding other means such as textual or graphical annotations to enable indirect contributions.

Given the increasing usage of graphical representations in organizations, their collaborative use and creation is of vital interest not only 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, but also for other disciplines.

The content of the papers in this volume point to interesting directions of research and presents cutting edge insights into the collaborative usage and development of models and other graphical representations in modern organizations. Thus, we are convinced it will be interesting for many different researchers and practitioners from several disciplines. We are also convinced that it provides a fertile ground for further research.

# References

Baacke, L., P. Rohner, R. Winter and R. Fitterer (2009). Component-Based Distributed Modeling of Collaborative Service Processes - A Methodology for the Identification of Reference Process Building Blocks: *Forty-Second Annual Hawaii International Conference on System Sciences (HICSS-42).*

Brosch, Petra, Martina Seidl, Konrad Wieland, Manuel Wimmer and Philip Langer (2009). We can work it out: Collaborative Conflict Resolution in Model Versioning. In Wagner, Ina; Balka, Ellen; Ciolfi, Luigina; Simone, Carla and Tellioæglu Hilda (eds): *ECSCW 2009: Proceedings of the 11th European Conference on Computer Supported Cooperative Work, 7-11 September 2009, Vienna, Austria.*

Heer, Jeffrey, Michael Bostock and Vadim Ogievetsky (2010). A tour through the visualization zoo. *Commun. ACM*, vol. 53. no. 6, pp. 59-67.

Herrmann, Thomas and Nolte, Alexander (2010). The Integration of Collaborative Process Modeling and Electronic Brainstorming in Co-Located Meetings. In Kolfschoten, G.; Herrmann T.; Lukosch S. (eds): *CRIWG 2010*. Berlin Heidelberg: Springer, pp. 145–160.

Klebl, Michael, Monika Hackel and Stephan Lukosch (2009). The Application of Design Patterns for the Adaptation of a Modeling Tool in Collaborative Engineering. In Carriço, Luís; Baloian, Nelson; Fonseca, Benjamim (eds): *Groupware: Design, Implementation, and Use. 15th International Workshop, CRIWG 2009, LNCS 5784*. Springer Berlin / Heidelberg, pp. 262–269.

Morris, M.; Ryall, K.; Shen, C.; Forlines, C. & Vernier, F. (2004), Beyond social protocols: Multi-user coordination policies for co-located groupware, *in* 'Proceedings of the 2004 ACM conference on Computer supported cooperative work', pp. 265.

Prilla, Michael & Nolte, Alexander (2010), Fostering self-direction in participatory process design, *in* Keld Bodker; Tone Bratteteig; Daria Loi & Toni Robertson, ed.,'Proceedings of the eleventh conference on Participatory Design 2010', ACM New York, , pp. 227--230.

Prilla, Michael (2010). Models, Social Tagging and Knowledge Management – A fruitful Combination for Process Improvement. In Rinderle-Ma, Stefanie; Sadiq, Shazia; Leymann, Frank (eds): *BPM International Workshops 2009, Ulm, Germany*. Ulm, Germany:

Renger, M.; Kolfschoten, G. & De Vreede, G. (2008), 'Challenges in collaborative modelling: a literature review and research agenda', *International Journal of Simulation and Process Modelling* **4**(3), 248--263.

Rittgen, Peter (2010), 'Collaborative Modeling: Roles, Activities and Team Organization', *International Journal of Information System Modeling and Design (IJISMD)* **1**(3), 1--19.

Yuille, Jeremy and Hugh Macdonald (2010). FEATURE The social life of visualization. *interactions*, vol. 17. no. 1, pp. 28-31.

In: Nolte, A.; Prilla, M.; Lukosch, S.; Kolfschoten, G. and Herrmann, T.: Proceedings of the 1st International Workshop on Collaborative Usage and Development of Models and Visualizations at the ECSCW 2011 (CollabViz 2011)

# An Approach for a Domain-spanning Collaboration Platform for Decision Support Using Immersive Visualization Techniques in Product Manufacturing

Daniel Eichhorn[1], Johannes Herter[2], Andreas Oberweis[1], Jivka Ovtcharova[2]
[1]Institute of Applied Informatics and Formal Description Methods (AIFB)
[2]Institute for Information Management in Engineering (IMI)
Karlsruhe Institute of Technology
Karlsruhe, Germany
*{daniel.eichhorn, johannes.herter, andreas.oberweis, jivka.ovtcharova}@kit.edu*

**Abstract** -This paper proposes an approach for cross domain collaboration in product development and manufacturing by linking domain specific information models and provides a common visualization by means of Virtual Reality technology. This approach should be considered as work-in-process in an early stage. It should contribute to a decision support platform in which manufacturing alternatives are evaluated regarding criteria of several domains. Therefore information models from process design and product engineering are accessed and linked via items they describe. Linked information models are displayed with stereoscopic visualization technology in order to provide an understandable and yet complete illustration. Users should be able to evaluate manufacturing alternatives and instantly see the impact of modifications in their native and other connected information models. Thereby domain-spanning impacts of manufacturing alternatives can be recognized beforehand and cross-domain communication can be accelerated and improved through mutual understanding of domain experts.

## INTRODUCTION

In recent years, manufacturers are addressed with a demand for individualized products leading to a rising number of product variations, which poses new challenges to construction and production processes in various industries. Besides, distinct national and international competition between manufacturers demand close collaboration between all disciplines of an enterprise involved in product and process development as well as value creation in production.

All representatives of the domains involved have their native view on product and process, which drives their priorities in the decision processes. In order to make Pareto-optimal decisions concerning all domains, all participants should be able to communicate their point of view and priorities and in best case be able to regard consequences of their actions in foreign domains. This holistic approach is followed in modern management strategies like balanced scorecard (Kaplan and Norton, 1992). The collaboration of multi-disciplinary teams requires a central, cross-domain information management system in order to have one ‘single point of truth’ for all necessary information to make strategic and operative decisions concerning product, process and value creation. Usually, this is implemented in

Product Lifecycle Management (PLM) systems, which claim to host all product related information throughout the full product lifecycle.

In order to provide decision makers and experts of a specific domain access to a multi-domain stock of information for their decision process in an intuitive and understandable way, we propose a Virtual Reality (VR) based visualization of product and process information to display the information from various domains simultaneously in an understandable and intuitive way.

The concept presented here does not aim to replace any inter-domain discussion; furthermore construction processes should not be automatized. The vision of this approach is to make interconnections between domains transparent to accelerate and ease the coordination between expert groups by extending the information available and visible to indicate junctions of several domains. This should lead to an acceleration of decision processes, improved quality of decision by an extended foundation and thereby benefits from cost reduction potentials. The paper is structured as follows: At first, we introduce the basic concept of our approach in detail. Afterwards we give a brief overview on related work and end with a summary and an outlook on our next steps.

# CONCEPT

This section covers fundamental paradigms of our concept. On this foundation, we describe the key elements, succeeded by an introduction of major benefits we aim to achieve. One driving factor for the illustration we propose, are the cognitive processes, which lead to domain specific information models. According to (Stachowiak, 1973) in his fundamental work about General Model Theory the driving factors in modeling are “Pragmatism”, “Reduction” and “Mapping”. The reduction is driven by pragmatism, which determines the selection of relevant elements to be included and eventually emphasized. This basic principle of information models in different domains results in specific models for process-monitoring, construction or economical analysis. In our scenario, all models deal with production processes but they regard different aspects of reality. The reduction of complexity of reality gives the possibility to make decisions from the point of view of a specific domain by regarding domain-specific models. The reductive characteristics of modeling pose risk for not regarding facets of reality sufficiently, which could lead to suboptimal decisions. Therefore, communication between representatives of domains with their specific models is required.

Here our approach is applied: We propose an integrated illustration of (information-) models from several domains in order to give domain experts an insight into interconnections between domain-specific models in order to ease the coordination between domains and create an improved mutual understanding between domain experts from several domains.

## Process Modeling

In the process modeling domain various notations have been proposed. We decided to use Petri Nets for describing and modeling processes for several reasons, although our approach can be applied to other modeling notations.

Petri Nets were chosen as they can be used for the specification and verification of processes (Adam et al., 1998). Furthermore they provide the benefit of a mathematical foundation, which makes them

suitable for process analysis and simulation. Petri Nets can deal with issues of manufacturing processes (Desrochers and Al-Jaar, 1995), and they are a quite well-known and well-researched process modeling technique (Reisig, 2010).
Formally, a Petri Net is a directed bipartite graph with two sets of nodes and a set of arcs. A node is either a place or a transition. In the graphical representation circles denote places and boxes denote transitions.
Petri Nets are described by the triple N = $(P, T, F)$, where P is the set of places, T the set of transitions and F $\subseteq (P \times T) \cup (T \times P)$ is a flow relation. The numerous proposed Petri Net variants can be subsumed in elementary and high-level Petri Nets.
For our approach the Petri Net must be capable to refer to information models of other domains. We chose high-level Petri Nets, because process objects in high-level Petri Nets consist of net elements described above and process constraints or performance indicators such as cost, time, roles, resources or place capacities. With these process objects we establish an interconnection from the process information model to information models from other domains. We define the process objects following the definition in (Betz et al., 2008). The resources are assigned to the transitions with the mapping function, *mapTranisitonToResources*,

mapTranisitonToResources: $T \rightarrow \mathcal{P}(Re)$ *where*

- $Re = \{re_1, \ldots, re_n\}$ is the set of all resources,
- $n$ is the number of resources,
- $T$ is the finite set of transitions and
- $\mathcal{P}(Re)$ is the power set of $Re$.

## Spatial Arrangement of Information

A major component of the approach described here is the integrated visualization of production related information.
The issue of linking information models from several domains has been a topic in research and industry for some time. Concerning product data, STEP (Standard for the Exchange of Product data) has been developed and standardized (PLMS, 2007). Initially we take links between information models for granted though we are aware of challenges when establishing interconnections. The vision to show users the interconnections of elements from several domains raises a major challenge: It might be technically possible to show elements from several domains conjointly, although the mass of information could overburden users with complexity. Hence, the presentation and interaction paradigms should support users to manage the amount of information.
There are several approaches for the visualization of huge numbers of data-elements (Jamieson and Alexandrov, 2007, Chen et al., 2007). In the present case, there is an additional issue of having information sources from several domains. This raises the challenge on the one hand of having a linked information network, on the other hand of having the native domains of the information elements still trackable. Therefore we propose a spatial arrangement of the domain-spanning information model in an immersive, stereoscopic environment. Here, an information space is established retaining the perspective of domains as virtual dimensions in the visualization environment.

Although a visualization of the information space is possible on classic two-dimensional screens, we focus on applying immersive environments as the analogy of domains to virtual dimensions should improve the understandability of the visualized information. The benefit of an immersive visualization compared to a display on a common screen, is evaluated within this approach in the near future. For a proof-of-concept application, which is currently under development, we use the immersive VR environment of LESC (Lifecycle Engineering Solutions Center) consisting of a three surface (front, side and ground) passive stereo projection, in combination with an optical tracking mechanism.

For the visualization of spatially arranged domain-spanning information elements, several prerequisites should be fulfilled. First of all, interconnected content from several domains should be available. Here we focus on process descriptions and associated resources. The visualization concept should be sufficiently generic to be extended on other domains. One way to arrange domain-specific information models in a virtual space would be a cylindrically orientated distribution as indicated in Fig. 1. Domain-model would be visualized on sections of a cylinder. Thereby users could have the area of interest, whether it is a specific domain model, an excerpt of a model or the interconnection between several models in focus by their current point of view. With physical motion, assuming technical conditions like in a Cave Automatic Virtual Environment (CAVE) with several projection screens, users could change their point of view and thereby put different aspects of the information space in focus.

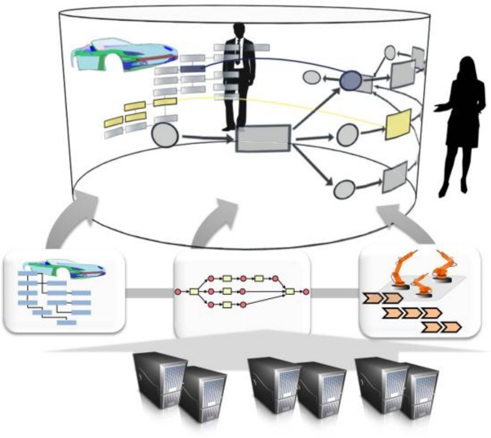

Figure 1. Concept overview

With this kind of information presentation we hope to achieve the following positive effects:

By putting certain aspects of illustrated information in foreground, users still have a reduced mapping of reality to work with, according to general model theory. The disadvantage of fully reduced information can be decreased, as several information models are displayed together without interfering each other fully. The contradiction of understandable presentation and reduction for dealing with complexity can be solved by an immersive presentation. The spatial distribution of models should help users to detect interconnection without being overstrained, as the perspective visualization puts information in background without full reduction. Thereby the user interface should be more understandable as the spatial orientation of humans is followed in the presentation. Users can intuitively alter the domain in focus, by changing their position instead of changing windows or having to use overloaded user-interfaces, like in common applications.

# Related Work

In (Betz et al., 2008) a Petri Net model with 3D components, enhanced with parts of an organization model (roles and resources), was proposed. The approach presented here extends this proposal. Elements of the resource view are used to link the process model to elements of the construction domain model.

Connections between different product manufacturing focused models to support several domain experts and their communication has been suggested in (Horváth and Rudas, 2009, Stanev et al., 2009). In

(Horvarth and Rudas, 1998), it was proposed to use Petri Nets for manufacturing processes. To show the impact between the several information models a data exchange between the models is needed. Several papers are addressing this problem (Bianconi, et al., 2006, Wang et al.,1989) on the base of existing data exchange formats.

Interoperability issues between information models of multiple domains, belong to the major challenges in research and industry. As described above, several approaches have been proposed to integrate information models from several domains. State of the art PLM solutions, which provide integrated information focus on data management, whereas the visualization of product and process data follows traditional approaches. Commercial PLM solution vendors apply new technologies for information visualization like VR techniques mainly on the visualization of geometry data like construction models or simulation data (visenso, 2010, IC:IDO VDP, 2010). The visualization of product related metadata is usually not tackled. VR solution providers focus on visualizing geometric information. Generally their products access mostly file-based information in CAx and PLM applications for a VR based visualization.

In general, these approaches with their benefits and disadvantages provide the possibility to apply VR in product manufacturing, though when regarded critically, they do not provide any additional information, which cannot be seen and analyzed in a two-dimensional representation. One critical matter of VR applications in this subject is the overall benefit they provide, meaning what additional information and improvement is provided to justify the price and effort of applying VR, as the content, which is already available and accessible in PLM and CAx systems is visualized in another user interface.

Regarding abstract information visualization in contrast to geometry-based data, there are some applications and paradigms followed. In general, these approaches can be classified as Visual Datamining applications. These applications use human ability for visual pattern recognition to find correlations when computer based algorithms are not applicable.

Some efforts were made to integrate PLM information in VR-based product visualization (IC:IDO VDP, 2010, Choi et al., 2009). These approaches focus on integrating information from several domains, though the focus lies on having some added information to the geometrical illustration of a product or a manufacturing resource.

# CONCLUSION

In this paper, we have introduced an approach for an integrated, context-sensitive visualization of Product-Process-Resource (PPR) data applying VR technology. This approach aims to improve mutual understanding between domain experts within the scope of product development. Information models, following the concepts of general model theory are derived for certain tasks and domains. This is in contrast to the demand for a holistic optimization regarding all perspectives in product engineering. For resolving collaboration issues by creating mutual understanding about reality, which is abstracted to models, we propose to use stereoscopic visualization techniques to map the domain-specific perspectives to spatial perspectives in the user interface. Thereby we assume to provide a more understandable and yet more complete illustration of product and production related information models. Our approach is considered to be work-in-process and is introduced to the research community for discussion about general suitability.

The approach presented here puts the process perspective in primary focus. For modeling processes, Petri Nets are used because of their mathematical foundation. Elements in the Petri Net are connected to technical product data without geometric representation, which interconnects the process perspective and product perspective. Currently a proof-of-concept application is under development focusing on product-process information visualization, though the general approach could also be extended to further application domains like Finance or Supply Chain Management.

# References

Adam, N. R., Atluri, V., and Huang, W. (1998): “Modeling and Analysis of Workflows Using Petri Nets”, Journal of Intelligent Information Systems, vol. 10 issues 2, Kluwer Academic Publishers Hingham, MA, USA , March/April 1998, pp. 131-158.

Betz, S., Eichhorn, D., Hickl, S., Klink, S., Koschmider, A., Li, Y. Oberweis, A., Trunko, R.: “3D Representation of Business Process Models”, Proc. Modellierung betrieblicher Informationssysteme, LNI P-141, Gesellschaft für Informatik, Bonn, Nov. 2008, pp. 73-88.

Bianconi, F., Conti, P., and Di Angelo, L. (2006): “Interoperability among CAD/CAM/CAE systems: A Review of Current Research Trends”, Proc. International Conference on Geometric Modeling and Imaging, London, England, July 2006, pp. 82-89.

Chen, J., Zheng, T., Thorne, W., Zaiane, O. R., and Goebel, R. (2007): ”Visual Data Mining of Web Navigational Data”, Proc. 11th International Conference Information Visualization, Zurich, Switzerland, July 2007, pp. 649-656.

Choi, S. S., Herter, J., Bruening, A., and Noh, S. D. (2009): “MEMPHIS: New Framework for Realistic Virtual Engineering”, Concurrent Engineering 2009, pp. 17-21.

Desrochers, A. and Al-Jaar, R. (1999): “Applications of Petri Nets in Manufacturing Systems; Modeling, Control, and Performance Analysis” Transactions on Semiconductor Manufacturing, IEEE Press, Dec. 1995, pp. 333-357.

Horváth, L. and Rudas, I. J. (1998): “Evaluation of Petri Net Process Model Representation as a Tool of Virtual Manufacturing”, Proc. IEEE International Conference on Systems, Man and Cybernetics, Piscataway, NJ, USA, Oct. 1998, pp. 178-183.

Horváth, L. and Rudas, I. J. (2009): “Indirect Communication between Engineers through Product Model”, Proc. 7th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, Sept. 2009, pp. 383-389.

IC:IDO VDP: www.icido.de, 05.10.2010

Jamieson, R. and Alexandrov, V. (2007): “A Data Forest: Multi-Dimensional Visualization”, Proc. 11th International Conference Information Visualization, Zurich, Switzerland, July 2007, pp. 293-300.

Kaplan, R. and Norton, D. (1992): “The Balanced Scorecard - Measures that Drive Performance”, Harvard Business Review, Jan.-Feb.1992, pp. 71-79.

Product Lifecycle Management Services V2.0, Object Management Group, 2007.

Reisig, W. (2010): “Petrinetze: Modellierungstechnik, Analysemethode, Fallstudien“, Vieweg + Teubner, Berlin, 2010.

Stachowiak, H. (1973): “Allgemeine Modelltheorie“, Wien 1973.

Stanev, S., Walla, W., Awad, R., Bittel, V., and Ovtcharova, J. (2009): ”Lifecycle Oriented Information Model to Support the Production Driven Product Validation.”, Proc. 6th CIRP-Sponsored International Conference on Digital Enterprise Technology, Springer Berlin/ Heidelberg, Dec. 2009, pp. 277-288.

Wang, F. Y., Gildea, K., and Rubenstein, A. (1989) “A COLORED PETRI NET MODEL FOR CONNECTION MANAGEMENT SERVICES IN MMS”, ACM Special Interest Group on Data Communication Computer Communication Review, New York, USA, July 1989, pp. 76-89.

www.visenso.de, 28.09.2010.

In: Nolte, A.; Prilla, M.; Lukosch, S.; Kolfschoten, G. and Herrmann, T.: Proceedings of the 1st International Workshop on Collaborative Usage and Development of Models and Visualizations at the ECSCW 2011 (CollabViz 2011)

# Towards Combining ThinkLets and Dialogue Games in Collaborative Modeling: an Explorative Case

S.J.B.A. Hoppenbrouwers[1] and W. van Stokkum[2]
[1]Radboud University Nijmegen, the Netherlands
and CRP Henri Tudor, Luxemburg
stijnh@cs.ru.nl
[2]Everest B.V., the Netherlands
w.van.stokkum@everest.nl

**Abstract.** We present a next step in our ongoing effort to conceive innovative support approaches for collaborative modeling. We propose to blend the well-established Collaboration Engineering approach (rooted in CSCW) with the Dialogue Game approach (rooted in Conceptual Modeling), viewing the second as a specialized extension of the first, and describing how they can complement each other. We hope to eventually link not only the approaches, but also the two fields. We provide a small but realistic illustration of our proposal at the hand of a real, industrially used elicitation pattern from knowledge modeling, and briefly show how this pattern can be wrapped up as an 'm-thinkLet'.

## Introduction

In many uses of collaborative modeling, e.g. in business engineering (den Hengst & de Vreede, 2004), knowledge engineering (Hoppenbrouwers, Schotten, & Lucas, 2010), problem structuring (Vennix, 1996), and enterprise engineering (Barjis, 2009), collaborative modeling with stakeholders untrained in modeling is a required and common practice, but also a continuous challenge, referred to as the 'knowledge acquisition bottleneck' (Hoppenbrouwers et al., 2010).

In the field of collaborative modeling (Renger, Kofschoten, & De Vreede, 2008), most work focuses on the collaborative creation and validation of model diagrams, using some standard modeling language (for example, UML activity diagrams: (Rittgen, 2007)). A different approach, which this paper is an exponent of, concerns more focused, 'smaller' conceptualizations that help gather and communicate highly to-the-point, well structured information that can be the basis for *derivation* (manually or possibly automatically) of more abstract, 'technical' models (Hoppenbrouwers, 2008; Hoppenbrouwers et al., 2010).

Once we move away from the 'collaborative diagram drawing' approach and into more limited and focused conceptualization (closer to the stakeholders' familiar concepts and requiring less skill in dealing with abstract syntax and complex visualizations and verbalizations), we can also move towards more closely guided, wizard-like conceptualization support (Hoppenbrouwers, Weigand, & Rouwette, 2009). We thus, in the long run, work towards the creation of a coherent library of well focused 'modeling games': rule-based, goal-driven interactive procedures that do not involve more than a few meta-concepts each and should be relatively easy to 'play' for stakeholders untrained in formal modeling (Wilmont, Brinkkemper, van de Weerd, & Hoppenbrouwers, 2010).

Such 'conceptualization games' bear considerable resemblance to the thinkLet concept central in Collaboration Engineering or CE (de Vreede & Briggs, 2005; Kolfschoten, Briggs, de Vreede, Jacobs, & Appelman, 2006), and can in fact be seen as a specialized extension of that approach. However, as will be explained in the next section, some additional properties are to be added to thinkLets as they (also) become Dialogue Games (DGs). The DG approach originated in the field of conceptual modeling, whereas CE concerns collaboration more in general, yet in the specific context of collaborative interaction support (in particular, CSCW). We hope to link not only the approaches, but ultimately also the two fields.

# ThinkLets and Dialogue Games

The Dialogue Game (DG) approach to collaborative modeling is rooted in a theoretical view on modeling as a *conversation* (Veldhuijzen van Zanten, Hoppenbrouwers, & Proper, 2004). Detailing this line of thinking led to a framework in which the core concepts are Rules, Interactions, and Models (RIM): **R**ules both drive and constrain conversational **I**nteractions that include propositions, but also argumentation *about* those propositions. A set of propositions as accepted by the modelers at some point in time constitutes a current **M**odel. For an elaborate explanation of the RIM framework, see (Ssebuggwawo, Hoppenbrouwers, & Proper, 2009). Interactions include conversational moves like arguing for or against a proposition, agreeing, disagreeing, and of course putting forward or withdrawing a proposition.

From the rule-based RIM approach, it is a small step to viewing modeling sessions as enacted games (instantiations of a game type). In addition, there is a theoretical link between the RIM approach and '*dialogue games*', a known concept in Argumentation Theory (Eemeren et al., 1996).

Let us now consider the CE approach (involving thinkLets) and see how this approach relates to the DG approach to collaborative modeling. Please note that lack of space prevents us from providing a full scale, detailed comparison between the CE and DG approaches here; we intend to do this elsewhere, including identification of overlap between existing thinkLets and (parts of) Dialogue Games. Indeed we know such overlap exists. However, our strategy is to first focus on the creation of playable game implementations; analysis and (re)-use of generic patterns (thinkLets) in these games will have to come later.

The CE concepts we refer to below are based on (Kolfschoten et al., 2006). Symbolical of the overlap between the two approaches, we refer to 'm-thinkLets': a (still mostly fictional) class of thinkLets for use in collaborative modeling and compatible with the structure of DGs.

In (Kolfschoten et al., 2006), thinkLets are defined as "named, packaged facilitation techniques that create predictable, repeatable patterns of collaboration among people working towards a goal". In (Hoppenbrouwers et al., 2009), collaborative modeling is characterized as a "goal-driven interactive activity that requires freedom of action and decision within clearly set boundaries." Games are typically also such activities. A similar direction is suggested in (Kolfschoten et al., 2006)) by shifting from the use of complete and rather detailed, restrictive 'scripts' as part of specifying thinkLets, to defining *rules*. Though they do not explicitly refer to 'games', from the DG/RIM perspective even classic thinkLets *are* games, of a sort.

In dealing with the optimal trade-off between constraint and freedom in guiding interaction, much can be learned from game dynamics. In addition, taking the game metaphor seriously suggests some interesting possibilities: the use of advanced interfacing from gaming to make collaborative interaction more accessible and engaging; even the use of devices like score systems or local competition embedded in over-all collaboration (Hoppenbrouwers et al., 2009).

The DG approach recognizes the long term goal (also highly prominent in CE) of removing the facilitator as much as possible (disintermediation), yet it currently focuses on simplifying and structuring the facilitator's role rather than removing it. A DG for modeling is typically viewed as two entwined games with distinct sets of goals and rules: one (or more) for the stakeholder-participants, one for the facilitator-participants. Again this merges the notion of 'rules' with the notion of 'script', including the facilitator as a role in the game. Such a setup was successfully executed in a pilot DG for Group Model Building, transforming a script into a DG (Hoppenbrouwers & Rouwette, 2011).

In modeling (as opposed to generic collaboration), a key notion is that of a *meta model* or modeling language. Though this aspect is in principle covered by thinkLet design concepts, it could benefit from additional, further specialized views from the DG approach. The *pragmatic* focus of a DG (the intended use of the conceptualization it renders: its desired resulting contents) is driven by focus questions; its *semantic-syntactic* focus (the modeling language or conceptual format of the result) constrains the formulation of focused answers (Hoppenbrouwers & Wilmont, 2010). Small sets of meta concepts used in modeling can thus be deliberately introduced in m-thinkLets, aiding their pragmatic and semantic-syntactic focus.

CE uses the concept of "parameters" of thinkLets: content-specific variables, for example focus questions. One could view such parametrization as an important aspect of the development of m-thinkLets. However, the creation of m-thinkLets would involve the setting of parameters that would still be generic for a certain flavor of modeling, e.g. ontological modeling, process modeling, and so on. Indeed, m-thinkLets require a specific, focused approach to the use of parameters extending into 'syntax setting' for m-thinkLet results.

CE covers 'moves of the game' that relate to the rendering of results of thinkLets. *Discussion* is explicitly included as a possible 'action' in thinkLets, but CE does not guide, constrain, or log its 'mechanics'. Contrarily, the DG approach considers the typical interactions of discussion and argumentation as discrete 'moves of the game' (Hoppenbrouwers & Rouwette, 2011). Logging all "discussion moves" and making them accessible both during and after the game is standard. Possibly, CE in general might benefit from such a mechanism.

Having explored key similarities and differences between CE and the DG approach, let us consider a realistic example of a potential DG based m-thinkLet.

# Example: The 'Weighted Factor Elicitation Game'

An exemplary 'm-thinkLet' interaction pattern was created in context of a project in which a radical new distributed model was conceived for scheduling Dutch railway traffic (van Stokkum, 1999). The pattern involved was applied in a one-and-a-half hour collaborative modeling session with three domain specialists of Dutch Railways, and a facilitator. A role playing setup was used to elicit the weighed factors that influence the creation of scheduling conflicts between trains.

The facilitator (a knowledge engineer) initiated the session by introducing a limited set of scenarios that can lead to a conflict. These scenarios were presented by schematic diagrams (Fig. 1).

The diamonds in Fig. 1 represent junctions. The other icons represent trains. The goal of the game is for the players to set parameters such that, for a specific scenario, there is a given p% chance (e.g. 75%) that the trains will raise a conflict (i.e. arrive at the same time) at the junction. During the game, the facilitator actively varies scenario details like the types of trains involved (e.g. length, load) or events occurring (e.g. wind conditions, engine failure ).


Figure 1. Two of the scenarios in which two trains could be arriving too close together at one infrastructural railtrack point

For example: “let domain expert 1 be the red train. This red train is a long cargo train carrying a heavy load. Domain expert 2 is the blue train which is IC train with high priority. Domain expert 3 is a junction that will assess continuously the chance of collision. Assignment: for this situation, collaboratively conceive and set factors so that there is a 75% chance the trains collide”. The actual, utilitarian goal of the game is to collaboratively define a stable set of factors influencing the chances of collisions taking place. Factor types thus elicited included *speed*, *maintenance record*, *weather influences*, *weight*, *type of engine*, *priority of passengers* and *cargo*; weights (high/low) indicated the importance of the factors.

The domain experts involved had no experience in creating formal models. The described session was one in a series of nine interrelated sessions, each of a similar focused nature. In each session the focus (both pragmatic and semantic-syntactic) was set differently to address a specific aspect: the train, the infrastructural points, creating conflict, creating a plan to prevent a conflict, determining a cost function to evaluate a plan, decision making on plans, determining follow-up conflicts, define a stop criterion for evaluating uncertain follow-up conflicts. By breaking up the problem into small, focused sessions, in the end a very complex distributed scheduling system was collaboratively modeled, without any ‘comprehensive diagram drawing’ (in fact, such a diagram would have too complex to draw in the first place: it was represented as a set of mathematical formulae).

The same patterns have later been reapplied in other projects in need of a real time distributed workflow scheduling solution. For example, the patterns have been used to develop a system for scheduling ground operations at Zaventem airport, scheduling autonomous operating robots in Rotterdam’s largest container handling terminal port, creating simulations to solve traffic jam problems in Holland and for developing an order picking system for distribution centers of a Dutch super market chain.

The technique presented above is an excellent example of a ‘Focused Conceptualization’ or ‘FoCon’ as introduced in (Hoppenbrouwers & Wilmont, 2010). Specifications of FoCons are somewhat similar to conceptual designs for thinkLets, but they were developed strictly in context of collaborative modeling.

FoCon analysis as an instrument concerns questions like: "What goes into a FoCon situation, in terms of existing information and people (including their concerns, knowledge, and skills)"; "What is the intended output of a FoCon situation, in terms of pragmatic goals, conceptual (semantic-syntactic) constraints set, and the required level and sort of agreement between people", "what focus questions are used, and what explicit instructions are to be given by the facilitator, in which situation", and "what rules govern the required or limited interaction between players, in view of a current focus question". Clearly, a similar analysis could be applicable in a thinkLet context. The main points of a FoCon analysis of the m-thinklet described above are given in Table I below:

| | | |
|---|---|---|
| **"IN"** | Info | Various given scenarios and given chances of collision |
| | Concepts | Trains, junctions, situations (diagrams); properties of trains, partly based on results of ongoing elicitation; given chance of collision (P-value, e.g. 0.75) |
| | People | Train traffic management experts, not trained in formal modeling, some system thinking ability, homogeneous professional background |
| **"OUT"** | Info (pragm. focus) | List, generalized over all scenarios used, of weighted factors influencing collision risk |
| | Concepts (sem.-synt. focus) | Factor types, weight for each factor type (high/low impact) |
| | Social req. | Factors commonly understood and agreed upon |
| | Argumentation | Arguments raised and accepted/rejected in discussing the factors and their weights |
| **Substeps/ Strategy** | | *Facilitator*: iteratively set scenario, then discuss factors, then change details of scenario or set new scenario, thus systematically exploring all factors and developing a generic overview; *Players*: assume role of train or junction; for a series of scenarios, provide weighted factors matching a given chance of collision |
| **Interaction Modes in the game** | | • Focus on shared understanding of scenario<br>• Focus on identifying relevant factors<br>• Focus on determining the weight of a factor |

Table I. Overview of the main points of a FoCon analysis and DG outline of the example

We hope the table sufficiently illustrates how a FoCon analysis can serve as a basis for designing both Dialogue Games and m-thinkLets. Note that in the example, 'argumentation' plays a role in the actual elicitation process (arguments can be looked up during a running game and are a source of ideas about factors for the players) but argumentation is also logged for future reference to details in the discussion (otherwise lost). Structure is inherently provided by the DG setup.

We leave out considerations of mappings between m-thinklets, aptly called "transitions" (Kolfschoten et al., 2006), except by stating that such transitions can be direct *mappings* of resulting concepts to models or model views, but also *derivations* (typically by means of logical reasoning) based on concepts found and possibly leading to further *abstraction* thereof (Hoppenbrouwers et al., 2010).

## Acknowledgments

This paper results from the Agile Service Development project (*http://www.novay.nl/okb/projects/agile-service-development/7628*), a collaborative research initiative focused on methods, techniques and tools for the agile development of business services. The project consortium consists of BeInformed, BiZZdesign, CRP Henri Tudor, Everest, HU University of Applied Sciences Utrecht, IBM, Novay, O&i, PGGM, RuleManagement Group, Radboud University Nijmegen, Twente University, Utrecht University, and Voogd & Voogd. The project is part of the program Service Innovation & ICT of the Dutch Ministry of Economic Affairs.

## References

Barjis, J. (2009). Collaborative, Participative and Interactive Enterprise Modeling. In J. Filipe & J. Cordeiro (Eds.), *Enterprise Information Systems*. Berlin, Heidelberg: Springer Verlag.

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.

den Hengst, M., & de Vreede, G. J. (2004). Collaborative Business Engineering: a Decade of Lessons from the Field. *Journal of Management Information Systems, 20*(4), 85--113.

Eemeren, F. H. v., Grootendorst, R., Snoeck Henkemans, F., Blair, J. A., Johnson, R. H., Krabbe, E. C. W., et al. (1996). *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Hoppenbrouwers, S. (2008). Community-based ICT Development as a Multi-Player Game. In C. Benoit-Barné, B. H. Brummans, F. Cooren, H. Giroux, A. Létourneau, D. Raymond & D. Robichaud (Eds.), *What is an organization? Materiality, agency, and discourse: a tribute to the work of James R. Taylor*. Montreal: Dept. of Organizational Communication, University of Montreal.

Hoppenbrouwers, S., & Rouwette, E. (2011). A Dialogue Game for Analysing Group Model Building: Framing Collaborative Modelling and its Facilitation. To appear. *International Journal of Organisational Design and Engineering, special issue on Collaborative Modeling*.

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.

Hoppenbrouwers, S., Weigand, H., & Rouwette, E. (2009). Setting Rules of Play for Collaborative Modeling. *International Journal of e-Collaboration (IJeC), 5*(4), 37-52.

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.

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.

Renger, M., Kofschoten, G., & De Vreede, G. (2008). Challenges in collaborative modelling: a literature review and research agenda. *International Journal of Simulation and Process Modelling, 4*(3), 248--263.

Rittgen, P. (2007). Negotiating Models. In J. Krogstie, A. Opdahl & G. Sindre (Eds.), *Advanced Information Systems Engineering, 19th International Conference, CAiSE 2007,* (pp. 561-573). Berlin, Heidelberg: Springer.

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.

van Stokkum, W. (1999). A Model for Distributed Multi-agent Traffic Control. *Multiple Approaches to Intelligent Systems* (pp. 480--489).

Veldhuijzen van Zanten, G., Hoppenbrouwers, S., & Proper, H. A. (2004). System Development as a Rational Communicative Process. *Journal of Systemics, Cybernetics and Informatics, 2*(4).

Vennix, J. A. M. (1996). *Group model building. Facilitating team learning using system dynamics*. Chichester: Wiley.

Wilmont, I., Brinkkemper, S., van de Weerd, I., & Hoppenbrouwers, S. (2010). Exploring Intuitive Modelling Behaviour. In I. Bider, T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt & R. Ukor (Eds.), *Enterprise, Business-Process and Information Systems Modeling. Proceedings of the 11th International Workshop, BPMDS 2010, and 15th International Conference, EMMSAD 2010, held at CAiSE 2010, Hammamet, Tunisia, June 7-8, 2010* (pp. 182--194). Berlin, Heidelberg: Springer Verlag.

In: Nolte, A.; Prilla, M.; Lukosch, S.; Kolfschoten, G. and Herrmann, T.: Proceedings of the 1st International Workshop on Collaborative Usage and Development of Models and Visualizations at the ECSCW 2011 (CollabViz 2011)

# Collaborative diagram drawing: a case study on scaffolding self-regulated behaviors

Ilaria Manno, Giuseppina Palmieri, Vittorio Scarano
ISISLab, Dipartimento di Informatica, Università di Salerno, Italy
*manno@dia.unisa.it, palmieri@dia.unisa.it, vitsca@dia.unisa.it*

**Abstract.**

In this paper we present a case study of collaborative diagram drawing involving 36 students in Computer Science. Their task was to collaboratively draw a Use Case Diagram about the scenarios provided at the begin of the experiment. As students of a Software Engineering course, they had a general knowledge of such type of diagrams and related terminology, but they were not experts and had not real and practical experiences in diagram drawing. The tools used were a synchronous collaborative drawing tool integrated with a chat tool to support communication among the participants. Moreover, the experiment has been structured following the ’think, pair, share’ method. The analysis of the collaboration process outlines a twofold result: first, a significant equal participation of all the students and second, an implicit and recurrent self-regulatory behavior employed by the students to create and refine the diagram and to reach agreement about the final result.

## 1 Introduction

The collaborative creation of diagrams is commonly used in brainstorming processes, development of models, problem solving processes and, in general terms, in the creation of shared knowledge and understanding. This activity is used also in the educational setting to support collaborative learning. Indeed, in educational settings the aims are different from the working setting: students are responsible for one another’s learning as well as their own, thus, the success of one student helps other students to be successful. Therefore, the aspects about the users’participation and the self-regulated behaviors become particularly important: the free-riding behavior of some students de-motivate the other students and, then, the overall team

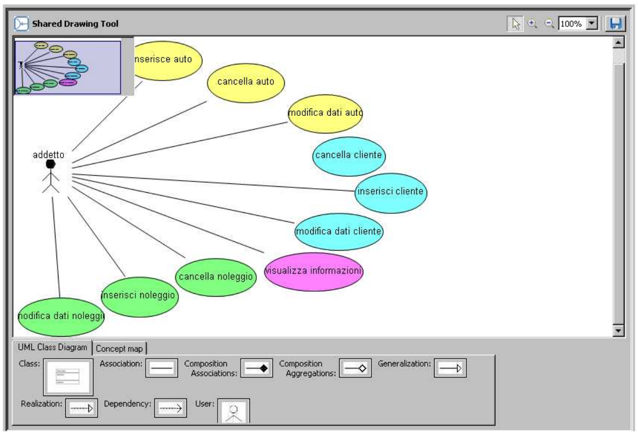


Figure 1. SDT screenshot showing the creation of a Use Case Diagram..

performance decreases (Ruël et al. (2003); Maldonado et al. (2007)).

In this paper we present our evaluation of an experiment of collaborative diagram drawing in a Software Engineering course. We have used a chat and a tool (the Shared Drawing Tool, SDT) for collaboratively drawing of Use Case Diagrams. Both the tools are integrated in the CoFFEE platform, which offers also the possibility to structure the collaboration in customized phases. In the analysis of the experiment we have found that all the students participated with a similar engagement and effort, and by using effectively the tools.

# 2 The experiment

Our experiment has been conducted in collaboration with the University of Basilicata (Erra et al. (2010)). It involved 36 students of the Software Engineering course; they had a general knowledge of such type of diagrams and related terminology, but they were not experts and had not real and practical experiences in diagram drawing. Their aim was to draw collaboratively a Use Case Diagram about some tasks proposed by the teacher. The experiment was designed so that half of the students worked in face-to-face (f2f) condition without the computer support and the other half worked in a (simulated) computer supported remote condition. Then, the two groups repeated the experiment in the opposite conditions. Erra et al. (2010) evaluated the diagrams and they found that, while the f2f setting needs less time to complete the work, the diagrams quality is slightly better in the computer supported condition. An early analysis of this experiment about students involvement focused on the reduction of the free-riding effect and found a significant equal participation among all the students (Belgiorno et al. (2010b)). Before presenting the data analysis, we briefly describe the software system used and the setting of the experiment.

The groupware used is CoFFEE (Collaborative Face-to-Face Educational Environment), a set of applications aiming to enhance the computer supported collaborative learning in f2f setting. In this experiment, we have simulated a remote condition: the students were in the same classroom but were grouped and seated so that they could not have f2f interactions. The main applications are the CoF-

FEE Controller and the CoFFEE Discusser, used in classroom respectively by the teacher and the students. These applications offer several collaborative tools which can be arranged together following the specific path designed preliminary by the teacher (De Chiara et al. (2007)). We have used the Shared Drawing Tool (SDT), a graphical tool integrated in CoFFEE to support the synchronous collaborative creation of graphs. The SDT offers a shared 2-dimensional space where the students can create figures and links, can move the existing items, and can edit his/her own contributions by changing the text, color, size, fonts (while editing contributions of other users is not allowed). The SDT provides also a direct support for creating concept maps and UML diagrams. A screenshot of the SDT is shown fig. 1.

### 2.1 The Experiment setting

The experiment involved 36 students in Computer Science of the University of Basilicata (27 Bachelor and 9 Master students). The assigned tasks were about a software system to manage *(a)* a library, *(b)* selling and rental of films, *(c)* a car rental, *(d)* an e-commerce platform to order CDs. The tasks were similar in complexity and were reasonable in relation with the preparation of the students. For each task the students were asked to provide a Use Case Diagram.

The collaboration through CoFFEE has been organized following the “Think, Pair, Share” method (TPS) to encourage students participation. The students were organized in groups of four people, and the activity was structured in three steps:
***think***, students work individually on the task to carry out; in this phase CoFFEE was configured to offer to each student his/her own istance of the SDT;
***pair***, students work in pairs on the task to carry out; in this phase CoFFEE was configured to manage groups of 2 persons and offer them the SDT and the Chat;
***share***, students work all together to produce a final solution; in this phase CoFFEE was configured to manage groups of 4 people and offer them the SDT and the Chat.

At each step, the results from the previous phase are copied onto the SDT workspace, so that the students can start the work of the new step on the basis of the previous one. The experiment generated 18 *traces*. A *trace* is an XML file where the Controller records all the events of a collaborative session: the chat messages, the shared drawing tool actions, clients connections and disconnections and so on. Two of the traces were corrupted so the data analysis is based on 16 traces.

## 3 Data analysis

A first study of the traces of the experiment aimed to evaluate the participation of students in the collaborative session by using the Gini coefficient (*Gc*) and it indicated that the participation among all the groups has been well balanced: the users who chatted more, drew less, and vice-versa, with no explicit agreement about the roles of the participants (Belgiorno et al. (2010b)).

We analyze, here, the experiment by looking at any pattern of coordination that can be found in traces. In general terms, the chat has been used as a mean to organize and coordinate the work on the diagram; moreover the usage of the chat and the contributions on the SDT are not totally casual: indeed, they present a regular

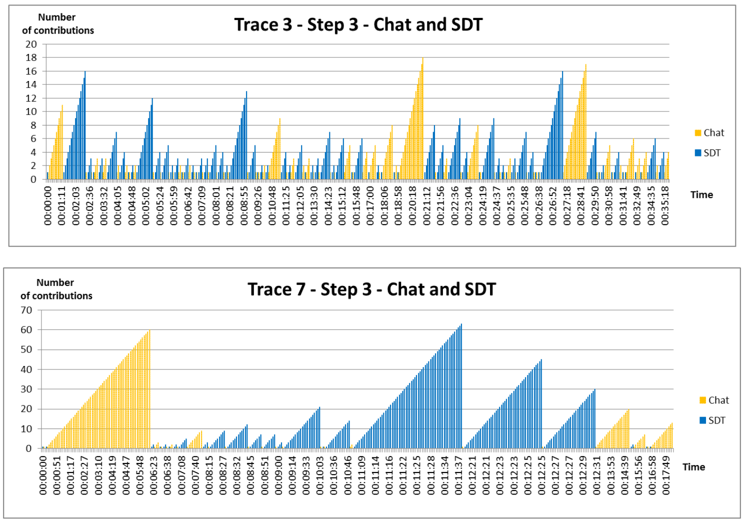


Figure 2. Above: the pattern A, with sequences of micro-phases involving chat and SDT. Bottom: the pattern B, with macro phases of chat and actions on the SDT.

pattern. In all the traces there is an initial phase in which the students use almost only the chat. This is due to the structure “think-pair-share” of the experiment: in the initial phase the students use the chat to describe the work that they have done in the previous step. After this initial phase, we found two kind of patterns, A (5 traces) and B (11 traces), which differ in the way of usage of the tools. In fig.2 we show the cumulative sequences of contributions on the chat and SDT.

The pattern A (top graphic of fig. 2) presents sequences of frequent chat messages followed by contributions on the SDT. In these traces, then, the coordination work goes through the whole phase as micro-coordination tasks.

The pattern B (bottom graphic of fig. 2) presents sequences of macro phases of many chat messages followed by macro phases of work on the diagram. In these traces, then, there are a well defined analysis and coordination phase followed by a wide phase of implementation of the work.

In the analysis of the patterns, we paid particular attention to the step 3 because it involves all the students in the work group, so we consider it as the most meaningful; however, most traces present the same kind of pattern (A or B) both in the steps 2 and 3 (we have not considered the pattern of the step 1 because, in the think-pair-share method, it is the step where each student *thinks* alone). It should be noticed that, in some traces, in the step 2, when the students are organized in two groups, one group presents the pattern A and the other group presents the pattern B; however, all of these traces present pattern B in the step 3. Therefore, the pattern B prevails on the pattern A and this could explain the greater number of traces of kind B. The students employed these pattern spontaneously: the teacher did not stimulate any behavior nor action, he was just responsible to pass from a step to the next one.

The patterns employed by the students seems to influence the level of re-using of the existing diagram through the several steps. As previously described, the structure of the experiment follows the “think, pair, share” method and in each step the

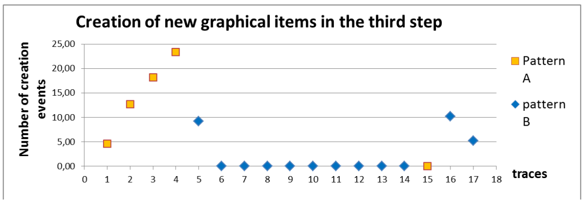


Figure 3. The traces with pattern B show few creation events on the diagram in the step 3, while the traces with pattern A show a meaningful higher number of creation events in the same phase.

students had the work of the previous step available as starting point. Then, the expectation is that at the third step most part of the diagram has been created and the students should work to organize and refine the final outcome by re-using the existing draw, with a minimal numer of creation of new items in the diagram. Indeed, the level of reusing of the existing diagrams and the number of new items created in the third step seem influenced by the pattern employed by the students: the groups adopting the pattern B create fewer items (mean 2,06% respect to the total actions on the SDT) than their colleagues employing the pattern A (mean 9,81% respect to the total actions on the SDT), as shown in fig. 3. This suggests that the students employing the pattern B, during the macro phases of work coordination, are able to optimize the work better than their colleagues and are able to achieve an higher level of re-using of the existing work.

# 4 Conclusions

In the collaborative usage and creation of diagrams it is fundamental supporting users' participation, reducing the free-riding effect and users' idleness as well as scaffolding self-regulated behaviors. We believe that a key factor is the integration of discussion and drawing tools in a seamless environment, so that the users can switch between the tools without any overhead, and the discussion and the drawing activities can converge in a single and natural collaborative flow. Moreover, we believe that, in the learning setting, it is fundamental the possibility to embed in the groupware a structure to drive the collaboration process: this allows to adopt well-known pedagogical strategies to enhance the students'engagement and learning performance. These ideas are supported by the analysis of the experiment that we have presented in this paper, which shows an equal participation of all the students and an effective usage of the tools to organize the work and create the diagrams.

We are aware that the conditions of our experiment (small groups, no facilitators or modellers, similar cultural background and modelling skills among participants) are very different from the business environment, where the modelling activity could involve larger groups, expert modellers, stakeholders with no modelling skills, and could require a severe check on the model quality (Renger et al. (2008)). Despite that, if the aim is to create a shared understanding among the participants, like in requirements elicitation or early phases of new projects, it could be more important supporting the participation and collaboration than a severe check of the model quality. In this direction it was oriented our work about the integration of collaborative tools in a software development environment (Belgiorno et al. (2010a)).

However, these cases require further analysis to evaluate if the creation of a shared understanding could be enhanced by an approach similar to our experiment: a system which integrates different tools could allow modelling skilled users to draw the diagram and no-skilled users to participate in the activity by using brainstorming tools. Moreover the participation could be supported by reducing the size of the groups through the management of subgroups, in order to reduce the necessity of a facilitator or a chauffeur. The convergence of subgroups through successive phases can have a twofold effect: the presentation of the ideas of each subgroup could highlight new sides of the treated problem and, at the same time, it is a double-check on the model to find and correct pitfalls (Frantzeskaki et al. (2008)).

**Acknowledgments.** The authors gratefully acknowledge the support by Andrea Guaiana during the development of Shared Drawing tool. The authors also thank the collaboration by Ugo Erra and Giuseppe Scanniello in setting up the experiment.

# References

Belgiorno, F., I. Manno, G. Palmieri, and V. Scarano (2010a): ‘Argumentation tools in a collaborative development environment’. In: *Proceedings of the 7th international conference on Cooperative design, visualization, and engineering*. Berlin, Heidelberg, pp. 39–46, Springer-Verlag.

Belgiorno, F., I. Manno, G. Palmieri, and V. Scarano (2010b): ‘Free Riding in Collaborative Diagrams Drawing’. In: M. Wolpers, P. Kirschner, M. Scheffel, S. Lindstaedt, and V. Dimitrova (eds.): *Sustaining TEL: From Innovation to Learning and Practice*, Vol. 6383 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pp. 457–463.

De Chiara, R., A. Di Matteo, Ilaria Manno, and V. Scarano (2007): ‘CoFFEE: Cooperative Face2Face Educational Environment’. In: *Proceedings of the 3rd International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007), November 12-15, 2007, New York, USA*.

Erra, U., A. Portnova, and G. Scanniello (2010): ‘Comparing two communication media in use case modeling: results from a controlled experiment’. In: *Proceedings of the International Symposium on Empirical Software Engineering and Measurement, ESEM 2010, 16-17 September 2010, Bolzano/Bozen, Italy*. ACM.

Frantzeskaki, N., A. De Haan, G. Kolfschoten, and W. Walker (2008): ‘Three challenges, four pitfalls and five actions when teaching conceptual modelling’. In: *International Conference of Education, Research and Innovation, ICERI 2008*. International Association of Technology, Education and Development (IATED).

Maldonado, H., B. Lee, S. R. Klemmer, and R. D. Pea (2007): ‘Patterns of collaboration in design courses: team dynamics affect technology appropriation, artifact creation, and course performance’. In: *CSCL’07: Proceedings of the 8th iternational conference on Computer supported collaborative learning*. pp. 490–499, International Society of the Learning Sciences.

Renger, M., G. L. Kolfschoten, and G.-J. De Vreede (2008): ‘Challenges in collaborative modelling: a literature review and research agenda’. *International Journal of Simulation and Process Modelling*, vol. 4, no. 3/4, pp. 248–263.

Ruël, G. C., N. Bastiaans, and A. Nauta (2003): ‘Free-riding and team performance in project education’. Technical report, Research Report 03A42, University of Groningen, Research Institute SOM (Systems, Organisations and Management).

In: Nolte, A.; Prilla, M.; Lukosch, S.; Kolfschoten, G. and Herrmann, T.: Proceedings of the 1st International Workshop on Collaborative Usage and Development of Models and Visualizations at the ECSCW 2011 (CollabViz 2011)

# Strategies in the Collaborative Use of Design Patterns

Claudia Iacob
Department of Computer Science, University of Milan
*iacob@dico.unimi.it*

**Abstract.** Originally proposed as a tool for knowledge representation and sharing addressing architects, the concept of design pattern has been adopted by other domains as well. This led to new and innovative ways of using it and its usefulness is largely recognized in the literature. However, little work has been done in investigating and measuring the impact a collection of patterns has on collaborative design processes involving designers. The paper describes the results of a case study involving 18 teams of undergraduate students in Computer Science. Making use of a collection of design patterns for the design of synchronous applications and being observed by a facilitator, they were asked to design applications which support synchronous collaboration. Abstracting from a) the sequences of actions the teams performed on the collection of patterns in isolated contexts of their design processes, b) the ratio of each category of actions the teams performed, and c) the facilitator's notes on the participants' interactions, a set of strategies the participants developed while using the patterns were identified and are presented in the paper.

## Introduction

Originally proposed as a tool for knowledge representation and sharing addressing architects (Alexander, 1977), the concept of design pattern – defined as "a proven solution to a recurring design problem" (Borchers, 2001) – has been adopted by other domains as well. This led to new and innovative ways of using it and its usefulness is largely recognized in the literature. On the one hand, software engineering applies design patterns for expressing Object-Oriented software design experience (Gamma, 1995). On the other hand, HCI designers adopted the design pattern approach to document and describe “the reasons for design decisions and the experience from past projects, to create a corporate memory of design knowledge” (Borchers, 2001; Schummer, 2007). In addition to that, patterns have been extensively used in teaching (Kolfschoten, 2010), bridging communication gaps between users and designers (Dearden, 2002), and abstracting results of ethnographic studies of cooperative work (Martin, 2002).

However, little work has been done in investigating and measuring the impact a collection of patterns has on collaborative design processes involving designers. This paper aims at providing some insight into the matter by describing a case study designed to answer the following question: “*What strategies do novice designers develop in working with a collection of design patterns?*” 18 design workshops were conducted with 18 teams of undergraduate students in Computer Science. They were provided with a collection of design patterns addressing the design of synchronous collaborative applications and were asked to use it in designing such an application. The patterns were identified through a 2-phase process fully described in (Iacob, 2011), comprising: 1) the analysis of the results of the design processes followed by 13 teams of designers, and 2) the analysis of 20 existing applications which support synchronous collaboration in activities such as drawing, text editing, searching, and games. The patterns included in the collection are briefly described below:

- **Who is the coordinator?** addresses the problem of providing a coordination mechanism which: a). allows all collaborators to take part in the collaboration and b). maintains the resource in a consistent state at all times.
- **Integrated chat** addresses the problem of supporting the communication among collaborators.
- **Eyes wide open** addresses the problem of allowing each collaborator to be notified about what the others are contributing to the process at any time.
- **Choose your collaborators** suggests allowing each user to be able to choose the people s/he wants to work with during the collaboration.
- **Collaboration, always social** suggests integrating social features in order to support the collaborators in forming a community.
- **With or without collaboration** addresses the issue of providing users with an additional private area, not available to the other collaborators.
- **My contribution** addresses the problem of supporting the identification of each individual’s contribution to the collaborative process.
- **Track history of collaboration** suggests saving the history of the collaborative process and making it available through repositories, or log files.
- **Adapt application to device** suggests supporting the materialization of the application on various devices.
- **Annotate** suggests allowing users to enhance the shared resource with textual, audio, or video notes on the misunderstandings they might have.

- **Support versioning** indicates enhancing the application with a versioning mechanism able to support the collaborators in viewing and editing older versions of the document they are working on.
- **Collaborative undo** suggests supporting the users in undoing changes performed on the shared document, maintaining the resource consistent.
- **Customize collaboration** points to providing the collaborators with the possibility of customizing the parameters of their collaborative process.
- **Shared summary** suggests providing the collaborators with an automatic way to create summaries of their collaborative processes.
- **Resume collaboration** suggests allowing the collaborators to pause their collaborative process, and restore it later.

# Case Study

This section presents a case study conducted for identifying the impact a collection of design patterns addressing the design of synchronous applications has on the collaborative design of such applications by novice software designers. 18 design workshops were conducted with 18 teams of undergraduate students in Computer Science. Making use of the patterns described above, they were asked to design the GUI and the interaction process of an application to support synchronous collaboration in activities such as drawing, text editing, game solving, and searching. Each pattern was represented on a paper card, being described by its *name*, its unique *ID*, the set of *keywords* associated to it, a representative *illustration*, the *problem* addressed by the pattern, and the *solution* proposed to tackle the problem.

The participants' design processes were audio recorded, a facilitator observed their interactions, and each participant provided his/her feedback on the workshop through a questionnaire. The recorded conversations of all the teams were transcribed. Their dialogues were divided into sentences (i.e. small fragments of dialogues – usually lines of the dialogues – related to a particular concept or action), all those sentences containing references to the patterns provided being filtered and considered for further analysis. The coding scheme used for coding the sentences referencing patterns classified these sentences as indicating: a) *browsing* the collection, b) *reading* a pattern, c) *using* a solution, d) *adapting* a pattern, e) *modifying* a pattern, f) *searching* for a pattern, g) *explaining* a pattern to another member of the team, h) *re-referencing* a pattern, and h) *generating* a design idea pointing to a pattern.

## Strategies in Collaborative Use of Design Patterns

Abstracting from a) the sequences of actions the teams performed on the collection of patterns in isolated contexts of their design processes (as defined through the coding scheme), b) the ratio of each category of actions the teams performed, and c) the facilitator's notes on the participants' interactions, a set of strategies the participants developed while using the design patterns were identified.

**Customize Pattern Identification**

In going through the patterns and trying to get familiar with the problems addressed by them, the teams often tried to associate each pattern with a characteristic word. Having done that, their dialogs would contain references to the patterns through the words associated to them (e.g. "*We can decide on a fixed time for all the game and during the game one can take maximum 2 breaks, and then we look into the solution for the pause one [the pattern Resume collaboration]*"). Interesting enough, these words were not consciously chosen from the list of keywords provided in the description of the patterns. However, with the exception of one case, all the words the teams associated with the patterns already belonged to the list of keywords provided by the cards.

Two of the teams filtered the collection of patterns after going through it and discussing it once and chose a subset of these patterns they considered fundamental for their design process. Throughout their work, they referred mostly to these patterns.

**Signal Patterns**

Often times, while some of the members of a team were focusing on the design task, the other(s) browsed the collection of patterns and tried to relate the team's design decisions to the solutions proposed by the patterns. When the team member(s) browsing the patterns identified a useful pattern at a specific moment, s/he signaled this pattern to the team. Some examples of such references are: "*Ok, there is a thing I read here [My contribution]: for understanding who has placed a certain piece*", or "*Look at this, this is interesting [points to pattern With or without collaboration] When you solve a puzzle you should have a private area where you try out the pieces and when a piece works well where it is placed, you just add it to the whole puzzle*".

**Search – Analyze - Apply**

The most common strategy the teams were expected to choose consisted in: a) initiate by writing down possible problems they would face, b) browse the collection of patterns searching for those patterns documenting the problems they considered, c) point to a pattern once found and read it, d) analyze the solutions proposed by the pattern and assess which solution to apply. Contrary to the expectations, less than half of the teams adopted this precise path of actions. However, all of the teams performed at least two of these actions during their design processes.

**Patterns as Checklists**

Eight out of the 18 teams used the collection of patterns also as a checklist. They initiated their work after going through the patterns, but initially ignored them. After reaching an idea for the application they were designing and sketching a draft of it, they went through all the patterns, one by one, in order to make sure that they covered all the issues addressed by the collection. For each of the patterns, they analyzed whether they considered the issue addressed by the pattern. In the affirmative case, they identified the solution they adopted. In the negative case, they explained the reasons for which the pattern did not apply to their design context.

**Patterns as Startup Tools**

Four of the teams initiated their design processes by going through the patterns, one by one, and identifying how could the pattern be applied in the context of their application's design. Then, when faced with a problem during their design process, the teams tried to remember which of the patterns addressed that problem. Examples of such references are: "Y*es, there was a pattern on that*", or "*There was one [pattern] that was mentioning the saving… because if we are 5 and we decide to save, we should be able to do that*". Moreover, specific situations faced during the design process reminded the teams of the patterns they browsed at the beginning of the process. As example of such a reference, consider "*Exactly, this was one of the issues in the patterns. If one clicks on the piece and drags it, in that moment that piece is locked*".

**Patterns as Source of Inspiration**

A common behavior of all the teams was to consult the patterns ever so often during their design processes. This helped them explore their design options and take informed decisions on the solutions to consider applying. Moreover, once going through the patterns, the teams would consider problems and design ideas they wouldn't have considered otherwise. Patterns inspired the teams in adding elements to their designs, and some example of references to such situations are: "*Let's add something about notifications [after reading Eyes wide open]*", or "*How do they choose the collaborators? [pointing to the pattern Choose your collaborators]*".

**Mark the Use**

The final result provided by each team was a sketch or a mockup of their overall design. No strategy was suggested to the participants for marking the patterns used. However, there were three ways they decided to address this. The majority of the teams grouped together all the patterns they used, putting them aside. Others have decided to arrange the patterns in the order they used them throughout the process. A more systematic approach was adopted by two of the teams which annotated their sketched with the IDs of the patterns they used, marking the use of each pattern in a specific context of the application's design.

**What do you mean?**

Patterns were often used as means of making oneself understood. The teams used the patterns in order to explain each other concepts or to discuss open issues or misunderstandings. For example, one of the most challenging concepts to grasp was reverting changes, the teams making use of the Collaborative undo pattern to explain each other the concept and the way it can be addressed in the context of the applications they were designing. Similar results have been identified in [3].

**Beyond Patterns**

During their work with the patterns, some of the teams went beyond the definition provided by the cards and pointed out examples of applications of the patterns in software systems commonly used. Moreover, one of the teams identified possible relationships existing between patterns. For example, they considered the patterns Track history of collaboration, Collaborative undo, and Support versioning related to each other, even if they did not specify exactly in which way these patterns are related. A similar association was identified among the patterns Collaboration, always social, Annotate, and Customize collaboration.

# Discussion and Conclusions

The strategies described above trigger a set of implications to the use of design patterns in collaborative design processes: a) Initiating by going through a problem-solution knowledge repository related to the design domain allows the designers to frame their ideas, and better understand the further implications of their early design decisions, b) As searching in such a repository is the most common action designers are expected to perform,  the representation of such a knowledge base should consider including a straightforward way of querying it, c) Using patterns collaboratively, designers should be able to signal patterns to one another, supporting them in sharing knowledge, d) Marking the use of the patterns directly on the design result (mockups, models) allows documenting design processes, supporting their review and understandability, e) A design pattern collection may be used as a checklist to support validating design results, models and decisions. As future work, professional software designers will be involved in such collaborative processes and their strategies and their feedback will be comparatively analyzed with those obtained from the current study.

# References


Alexander, C., Ishikawa, S., Silverstein, M. (1977) *A pattern language: Towns, buildings, construction*. New York: Oxford University Press.

Borchers, J. (2001) *A Pattern Approach to Interaction Design*. John Wiley & Sons, Inc.

Dearden, A., Finlay, J., Allgar, E.,  Mcmanus, B. (2002) Using Pattern Languages in Participatory Design. *Proceedings of PDC'02*. CPSR, Palo Alto, CA., 2002.

Gamma, E., R. Helm, R. Johnson, Vlissides, J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

Iacob, C. 2011. A Design Pattern Mining Method for Interaction Design. *Proceedings of EICS2011*, Pisa, Italy, June 13-16, 2011 pp. 217-222

Kolfschoten, G., Lukosch, S., Verbraeck, A., Valentin, E., de Vreedea, G.J. (2010) Cognitive learning efficiency through the use of design patterns in teaching. *Computers&Education*, 54(3), April 2010, pp. 652-660

Martin, D., Rouncefield, M., Sommerville, I. (2002) Applying patterns of cooperative interaction to work (re)design: e-government and planning. *Proceedings of CHI '02*, 235-242.

Schummer, T. and S. Lukosch. 2007. *Patterns for Computer-Mediated Interaction*. John Wiley & Sons, Ltd.

In: Nolte, A.; Prilla, M.; Lukosch, S.; Kolfschoten, G. and Herrmann, T.: Proceedings of the 1st International Workshop on Collaborative Usage and Development of Models and Visualizations at the ECSCW 2011 (CollabViz 2011)

# Fostering the usage of process models for supporting departments in organizations

Nina Claus
Institute for Information Management and Information Systems, University of Osnabrück, Germany
*nina.claus@uos.de*

**Abstract.** Modeling business processes in organizations tends to be cost-, time and resource-intensive. Therefore, it is surprising that the achieved models are mostly used in a limited way. Although, the supporting departments such as Knowledge Management, Quality Management and Change Management have access to the models, the rate of use is low. The reason might be that existing approaches in practice focus primarily on the requirements of process owners and do not address managers of the departments mentioned above. To overcome this, a method is offered in this paper that implies not only questioning the process owners but also the other managers. The results of the questionnaire will help to reveal the untapped potentials of the process models and facilitate to adapt the process models for both target groups.

## Introduction

Nowadays, graphical representations of business process models exist in many large organizations and the number is increasing (Hill et al., 2009). Usually, these process models are used for process improvement, quality control and the supporting information technology (Fettke, 2009). In most cases, however, the process owner determines which knowledge should be archived with process models. Other departments, such as Knowledge and Quality Management are usually not involved. These departments often cannot use the models since necessary information for them is missing. This poses an interesting research question: How can the usage of the existing process models be increased? This paper proposes a method in which the Delphi approach is applied in order to find out how the needs of every party can be met and their individual interests included. Thus, the knowledge of different experts will be disclosed, related contents merged, evaluated and modified over a defined number of rounds (Linstone and Turoff, 2002).

# Existing Approaches

In literature, various approaches concerning the usage of process models can be found. Kesari et al. (2003) examined advantages and disadvantages of business process modeling, asking eleven experienced consultants. Sedera et al. (2004) built a process modeling success model based on three case studies. Davis et al. (2006) used a web-based survey to analyze the usage of conceptual modeling in Australia (Fettke, 2009, in Germany). With the focus on process models as knowledge imparting artifacts, Prilla (2009) conducted qualitative expert interviews with six professionals from different sectors. Overall it can be stated that approaches so far have been limited to focusing on the managers of the business processes and respective modelers. The participation of stakeholders in the (collaborative) modeling process is taken up for example by Niehaves and Plattfaut (2011). However, the work focuses on the usage of existing models by different stakeholders (Weske, 2007), including the supporting departments.

# Conceptualization

The proposed method consists of three steps due to the usage of the Delphi approach (see Figure 1).

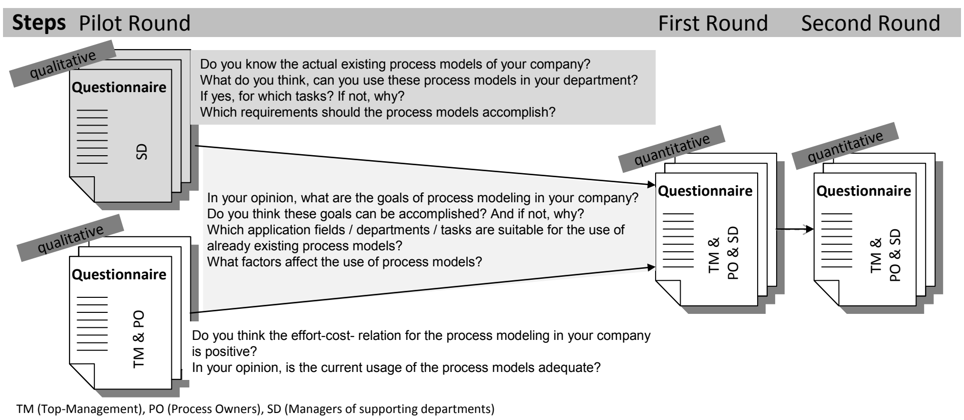


Figure 1. Procedure of the method.

The aim of this method is to incorporate the views of the Top-Management and Process Owners as well as the managers of the supporting departments. The figure also contains the most important aspect, i.e., the questions which should be asked. To ensure a shared understanding of both stakeholders, the results of the pilot round will be united and evaluated. The resulting extracted items will be combined, clarified and used for developing a structured questionnaire for the first round. Now, all participants should answer the same questions with the intention of consensus finding. After a quantitative analysis, results will be used to revise items for the second round or a third one; then the results should be concordant.

# Conclusion

The use of existing process models can be increased, if a wider range of users is involved from the very beginning. To ensure this, the method addresses how to gather the relevant data to identify potentials for an extended usage. Next steps will concentrate on identifying appropriate experts and conducting the interviews.

# References

Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S. (2006): 'How do practitioners use conceptual modeling in practice?' *Data & Knowledge Engineering,* vol. 58, no. 3, 2006, pp. 358–380.

Fettke, P. (2009): 'How Conceptual Modeling Is Used', *Communications of the Association for Information Systems*, vol. 25, no. 1, article 43, December 2009, pp. 571-592.

Hill, J. B., Lheureux, B. J., Olding, E., Plummer, D. C., Rosser, B., Sinur, J. (2009): *Predicts 2010: Business Process Management Will Expand Beyond Traditional Boundaries*, available on Gartner's website at http://www.gartner.com/resId=1231219.

Houy, C., Fettke, P. and Loos, P. (2010): 'Empirical research in business process management – analysis of an emerging field of research', *Business Process Management Journal*, vol. 16, no. 4, 2010, pp. 619-661.

Kesari, M., Chang, S., Seddon, P. (2003): 'A content-analytic study of the advantages and disadvantages of process modeling', *14th Australasian Conference on Information Systems*, 26-28 November 2003, Perth, Australia.

Linstone, H. A. and Turoff, M. (2002): *The Delphi Method, Techniques and Applications*.

Niehaves, B., Plattfaut, R. (2011): 'Collaborative business process management: status quo and quo vadis', *Business Process Management Journal*, vol. 17, no.3, 2011, pp. 384-402.

Prilla, M. (2009): 'Models, Social Tagging and Knowledge Management? A fruitful Combination for Process Improvement', *Proceedings of the 2nd Workshop on Business Process Management and Social Software*, 07 September 2009, Ulm, Germany.

Sedera, W., Gable, G. G., Rosemann, M., Smyth, R. W. (2004): A success model for business process modeling: findings from a multiple case study, *Eighth Pacific Asia Conference on Information Systems*, 8-11 July, Shanghai, China.

Weske, M. (2007): *Business Process Management: Concepts, Languages, Architectures,* Springer, Berlin and Heidelberg.

In: Nolte, A.; Prilla, M.; Lukosch, S.; Kolfschoten, G. and Herrmann, T.: Proceedings of the 1st International Workshop on Collaborative Usage and Development of Models and Visualizations at the ECSCW 2011 (CollabViz 2011)

# An Integrative Approach to Diagram-based Collaborative Brainstorming

Diogo Azevedo [a], Jordan Janeiro [b], Stephan Lukosch [b], Robert O. Briggs [c], Benjamim Fonseca [a]
[a] School of Science and Technology University of Trás-os-Montes e Alto Douro, [b] Systems Engineering Department Technological University of Delft, [c] MIS Department, San Diego State University, California, USA
*al33070@utad.eu*, *j.janeiro@tudelft.nl*, *s.g.lukosch@tudelft.nl*, *robertowenbriggs@gmail.com*, *benjaf@utad.pt*

**Abstract.** The need for computer supported collaboration has grown over the last years and made collaboration processes an important factor within organizations. This trend has resulted in the development of a variety of tools and technologies to support the various forms of collaboration. Many collaborative processes, e.g. strategy building, scenario analysis, root cause analysis and requirements engineering, require various collaboration support tools. Within these synchronous collaborative applications to create, evaluate, elaborate, discuss, and revise graphical models, e.g. data flow, fishbone and brainstorming diagrams, play an important role. Currently, the necessary tools are not integrated and flexible enough to support such processes. In this paper, we introduce a synchronous collaborative brainstorming diagram editor that is integrated in a flexible group support system. By this our approach goes beyond the current state of the art as we can be seamlessly integrated with other collaboration support tools such as text-based brainstorming, voting, etc.

## 1. Introduction

Working practices had an important growth over the years, especially on group works - a group of people engaged in the execution of several objectives of a common task (Rowley A., 2006, and Frost and Sullivan, 2007). Therefore such an effort should be helped by collaborative practices such as the Computer Support and Cooperative Work - CSCW, which improves the performance of a group in the execution of tasks, through group work supported by information and communication technologies. Groups can become even more productive when supported by Group Support Systems – GSS. It is decisive that GSS adopt techniques for the development of groupware applications that meet non-functional requirements (quality attributes) such as interoperability, integration, reliability and usability (Ana B. Pelegrina, et al, 2010).

Many collaborative processes, e.g. strategy building, scenario analysis, root cause analysis and requirements engineering, require various collaboration support tools. Within synchronous collaborative applications to create, evaluate, elaborate and revise graphical models by groups, e.g. data flow diagrams, work structure breakdowns and fishbone diagrams. Currently, there is lake of support on GSS for such processes. GSS must therefore offer users collaborative environments where they can interact (Rafael Duque, et al, 2009), however many of these systems fail when providing the right tools for effective collaboration (Grudin J., 1994). Analyze how groups work and evolve is necessary when we consider the social dimension of the collaborative work (Grudin J., 1988).

In this paper we present a Collaborative Line-and-Symbol Diagramming Component – CLSD Component which offers a collaborative environment to manage graphical models and thereby their related collaborative processes. To achieve such a collaborative environment we have been concerned with awareness that as claimed by Dourish and Bellotti (1992) is defined as an understanding of others activities, which provides a context for your own activity. According to (Carl Gutwin, et al, 2004) group awareness information includes knowledge about who is on the collaborative environment, where they are working, what are they doing and their further intentions (Ana B. Pelegrina, et al, 2010). Furthermore, we took into consideration which techniques and diagram types can be used to support collaborative diagramming efforts, and how the features and functions of a single-user differ from a multi-user diagramming tool in order to optimize the values that groups can create through collaborative diagramming. CLSD Component is integrated as a plug-in component within the Computer Assisted Collaboration Engineering (CACE), and thereby can be used in various different processes. CACE approach embeds collaboration expertise with collaboration technologies (Briggs, et al, 2010), so that participants can gain the same benefits without any special training (Mametjanov, et al, 2011).

In the remaining of this paper, we define a set of concepts required within GSS and for Collaboration purpose. After that, we present the requirement analysis giving a scenario of collaborative processes and thereby the set of requirements. In the next chapters the architecture, features and modeling of the CLSD Component are addressed. Before concluding this paper, we fully explain the approach used to implement the CLSD Component.

# 2. Group Support Systems

A Group Support Systems - GSS consists on a suite of tools for focusing and structuring discussion, while it reduces the cognitive costs of communication and information access among group members making a joint cognitive effort towards a common goal (Robert and Briggs, 2000). Under certain circumstances, industry, military and academic groups who use GSS were able to realize substantial gains in productivity (Fjermestad and Hiltz, 2000). However, the set of tools that GSS can offer are restricted with a limited set of configurable features, for example it can be difficult to fit a collaborative process in a GSS platform. Furthermore, GSS platforms must be flexible to be personalized according to the processes, but to accomplish it they need to follow the component-based software development concepts to become more suitable to different processes' parts. There is a new generation of groupware systems following the component-based approach, such as DACIA (Ladu and Parakash, 2000) to support mobile applications, CoCoWare (Slagter, et al, 2000) to develop applications, and TeamComponents (Jörg and Claus, 2000) to develop either single-user or groupware applications.

# 3. Requirements Analysis

In the following we elaborate on the requirements that a GSS system has to fulfill allowing collaboration engineers to configure synchronous collaborative applications that actually fit specific collaborative processes, such as strategy building, scenario analysis, root cause analysis and requirements engineering. To illustrate these requirements we present a scenario of a collaborative strategy building process that uses collaborative diagramming and other collaborative applications, e.g. a text-based brainstorming. Two activities that can be considered in this scenario are: 1- a text-based brainstorming for strategy building; 2- a diagram-based brainstorming to organize, connect and manage strategies based on the data gathered in the previous activity (Figure 2).

In the above scenario, we have to support collaboration engineers in (R1) designing collaborative processes, such as strategy building, root cause analysis, and (R2) design suitable collaboration support. For that the GSS needs to support (R3) the integration of components that support collaborative processes, by allowing re-using of existing components (Ana B. Pelegrina, et al, 2010). Furthermore, it must be able to (R4) share, exchange and efficiency (interoperability) of data between components (Hofte H., et al, 1995, and Simone C., et al, 1999, and Ana B. Pelegrina, et al, 2010), in order to re-use the data gathered for example from the first activity (text-based brainstorming) into the second activity (diagram-based brainstorming). Additionally, we do not know all the support that is needed so that (R4) the set of components must be extensible

(Ana B. Pelegrina, et al, 2010) by software developers and (R5) man API to support them (Riehle D., 2000) should be provided. Finally, (R6) our scenario requires collaborative diagramming, and for that we have identified additional requirements.

The list of requirements is based on the analysis of other existing Diagram Software, such as Banxia[1], Smart Ideas[2] and Ext Designer[3]. In this case, the requirements address the interaction that Collaborative Diagramming has to provide to groups while they participate in collaborative environments. It must be possible for group members to (R6.1) insert, import (text-based) and manage ideas into a diagram-based format, like our previous strategy building scenario. Following, ideas are (R6.2) diagram-based organized (clusters and color manager) and (R6.3) connect arrows (connect ideas through arrows). Lastly, group members can unintentionally provoke data conflicts between contributions and therefore it is required to provide (R6.4) feedthrough (Dix A., et al, 1993) - context awareness with the scope (who has been doing what) of other members' activities, consequential communication (Segal L., 1995) - data with their information and the resources that are nearby, and also (R6.5) trigged locking mechanisms when updates occur.

# 4. Approach

According to Ana B. Pelegrina, et al, 2010 there are GSS systems addressing some of the requirements described above, however for our approach we have chosen a GSS called Action Centers because it addresses all of the above requirements and it fits with our purpose. Two parts form Action Centers: a CACE editor and a Process Support System (PSS). The CACE editor is a tool to design an effective work practice by defining the content and sequence of collaborative activities that are packaged into the PSS (Mametjanov, et al, 2011).

The Action Center therefore does not have any tools, as alternative these tools are plugged into the Action Center as components to simply make them available in the running system. So, the Action Center supports the design of collaborative applications (R1), and allows components (as our CLSD Component) to be assembled by Collaboration Engineers into the CACE editor (R2). These components have access to shared data (R4), are configurable (R3) and can be (re)-designed by other Collaboration Engineers. They usually consist of a user

---

[1] Banxia (Decision Explorer) is a proven tool for managing software issues. Structure and analyse of qualitative information. More information can be found in http://www.banxia.com/dexplore/.

[2] Smart Ideas concept-mapping software brings the power of visual learning to classrooms, through interactive white boards. More information can be found in http://smarttech.com/.

[3] Ext Gui Designer is a graphical user interface builder for web applications. Developed by Sierk Hoeksma. More information can be found in http://www.projectspace.nl/.

interface for displaying data shared in a group, some input mechanism, and business logic.

Furthermore, the Action Center provides two Javascript objects to manage data and their updates – ActionCenterListener, and an ActionCentersAPI (R5) that offers services to create and support the development of collaborative components. Additionally, the data is managed through dynamic communication channels using CometD[4] to a Universal Data Model (Mametjanov, et al, 2011), to dynamically create and store arbitrary relational data. The UDM and the two JavaScript objects offer some mechanism to manage contribution, such as *modifiedBy* to know who (6.4) changed the data, and *lockedBy* to (R6.5) edit-lock entities and their attributes to provide single-user editing. A more detailed description of the system can be found on (Mametjanov, et al, 2011).

Action Center does not address all requirements needed for Collaborative Diagramming. For that purpose, we implemented our (R6) CLSD Component that consists of an XML wrapper and an implementation in JavaScript with Ext JS[5] and an extended library called Joint JS[6]. The JointJS library is used for (R6.1, R6.2 and R6.3) creating diagrams that can be fully interactive for both implementing a diagramming tool (as our CLSD Component) as well as simply for publishing diagrams.

The CLSD Component is a web-based application that supports the cooperation of group participants towards group work. For example, it might support the group in a text-based or a diagram-based brainstorming. Figure 1 shows the overall architecture of our approach.

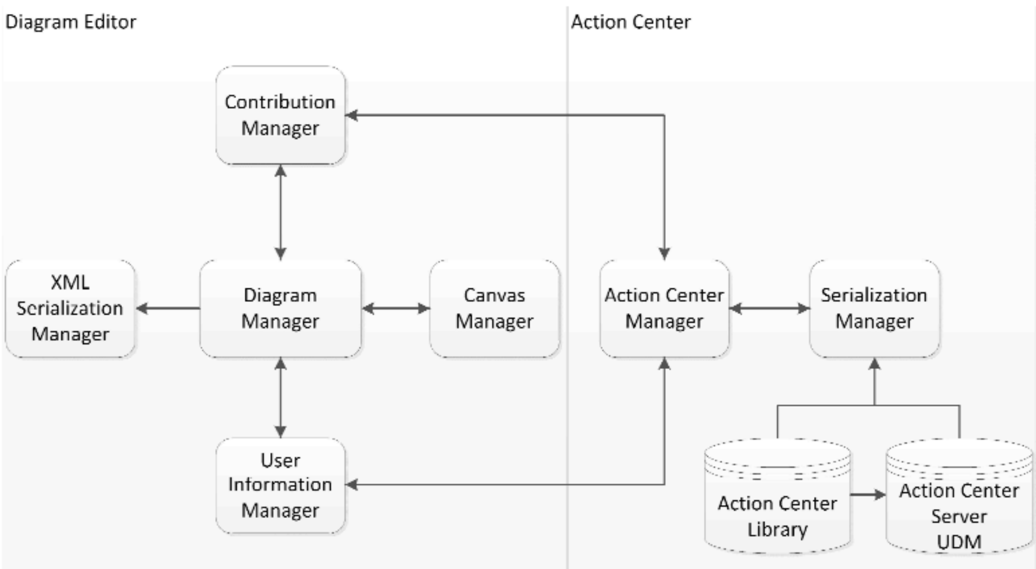


Figure 1. The CLSD Component Architecture coupled to Action Center

---

4 The Dojo foundation. Cometd. More information can be found in http://cometd.org/.

5 Ext JS is a javascript framework for developers. More information can be found in http://www.sencha.com/.

6 Joint JS is a JavaScript library developed by David Durman, More information can be found in http://www.jointjs.com/.

The Diagram Manager is the core manager of our CLSD Component, it is responsible for all processes of input and output and their distribution through the overall system, and for all connections inside the Diagram and between the Action Centers and the Diagram. Additionally, it connects with the Canvas Manager that is the bridge between the core manager of our system and the user - Figure 2. The User Interface (UI) influences its degree of acceptance since it allows communication, collaboration and coordination activities among several users interacting with the system (Victor M. R. P., et al, 2008). The Canvas Manager manages the CLSD Component design, the concepts and their connectors, and the collaborative tools / awareness mechanisms required (Carl Gutwin, et al, 2005), such as the list of users in the session, Telepointers – support actions, intentions and location awareness (Victor M. R. P., et al, 2008), and feedthrough – actions of a particular user can be shown to other users that collaborate in some task (Dix A., et al, 1998 and Carl Gutwin, et al, 2004).

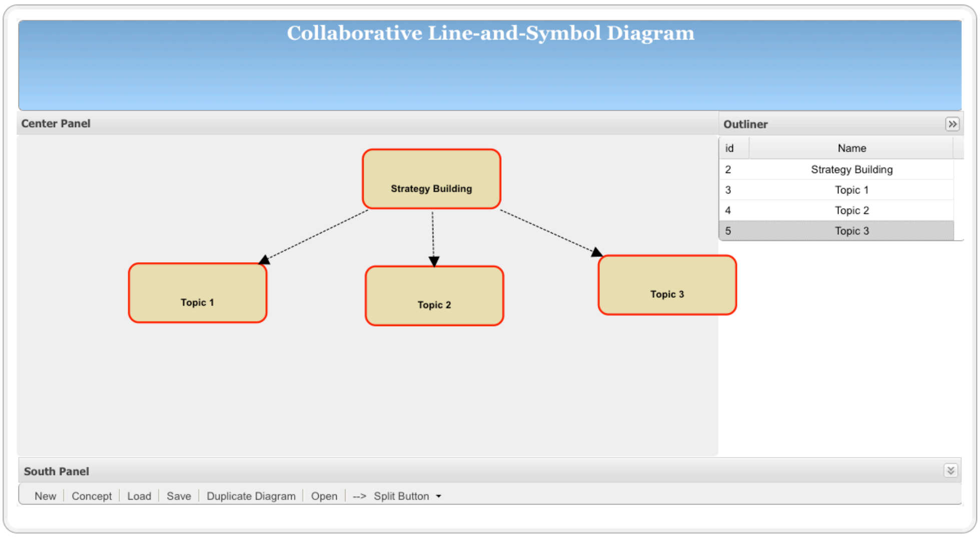


Figure 2. Collaborative Line-and-Symbol Diagramming Component

The Contribution Manager can also be called of Diagram Database Manager since it is responsible for adding, fetching and updating contributions to the Action Center Database. These contributions that are sent to the database can include concepts, arrows, JSON messages or objects and are trigged through notification mechanisms. To manage the information of users that are working in the diagram, such as listening online users, giving personalized information of each of them, and the scope of their activities - group awareness becomes a critical component in successful coordination (Carl Gutwin, et al, 2004) - we have implemented the entity User Information Manager. More information about groupware applications functionalities can be found at (Carl Gutwin, et al, 1998 and Carl Gutwin, et al, 2002).

Finally, another feature developed was the XML Serialization Manager, which is an output file that allows users to visualize their diagrams out of the Action Center.

Action Center in combination with CLSD allows us to support various different processes that require different forms of collaboration. Taking on consideration our previous scenario we take a closer look to the strategy building processes. Data is gathered from a text-based brainstorming (Outliner Component – First Activity) and stored into the UDM – Universal Data Model in Action Centers. The union of the Outliner Component with the CLSD Component (Second Activity) creates an Action Center, where the data, which is selected (identified) based on their relationship types and attributes by the Action Centers, is forward fetched (import) from the UDM and loaded (insert) to the CLSD Component. CLSD transforms it into a diagram-based format where group members can further manage and organize data as collaborative processes. Each single user controls the selection and manipulation of data and until he or she is finished no one else can have access to manipulate that specific data. For that purpose at each moment (through notification mechanisms) concepts shows a locking icon and a scope of action (feedthrough) of the user who is manipulating it.

# 5. Final Remarks and Future Work

Working practices can become even more productive when supported by GSS. They are becoming widely used thanks to the improvement of network infrastructure, communications, and development tools (Victor M. R. P., et al, 2008). Currently, the necessary collaboration support tools to create, evaluate, elaborate, discuss, and revise graphical models are not integrated or flexible enough (Akhil Mehra, et al, 2005) within GSS to support collaborative processes, such as data flow, fishbone and brainstorming diagrams. According to Bratitsis and Dimitracopoulou (Bratitsis and Dimitracopoulou, 2006), the techniques and information used by awareness mechanisms to the analysis of collaborative processes in which users accomplish common goals is considered the further step. For that there are several models to describe users actions in collaborative environments (Martínez et al, 2003).

In this article, we presented a Collaborative Line-and-Symbol Diagramming Component – CLSD Component assembled in a CACE editor to address the above challenges. A collaboration support tool that consists of a XML wrapper and an implementation for creating diagrams that can be fully interactive for both implementing a diagram-based brainstorming session to manage collaborative processes as well as simply for publishing diagrams. Furthermore, group members can insert, import (text-based) and manage ideas into a diagram-based format through a collaborative environment provided by the GSS system.

In future work, we will observe how practitioners and experts interact with our CLSD Component. We want to use the results to improve the flexibility and usability (Holzinger, 2005) of our component, and further see the exchange of data between components when changing from a text-based brainstorming to a diagram-based brainstorming.

# 6. Acknowledgments

The authors would like to acknowledge to all colleagues from University of Trás-os-Montes e Alto Douro - UTAD and Technological University of Delft - TU Delft that contribute to the development of our Collaborative Line-and-Symbol Diagramming Component.

This work has been partially supported by the FP7 EU Large-scale Integrating Project SMART VORTEX (Scalable Semantic Product Data Stream Management for Collaboration and Decision Making in Engineering) co-financed by the European Union. For more details, visit http://www.smartvortex.eu/

# 7. References

Grudin J., (1988): 'Why applications fail: problems in design and evaluation of organization or organizational interfaces', *Proceesdings of the ACM conference on computer-supported cooperative work*, pp. 85-93.

Dourish, P. and Bellotti, V. (1992): 'Awareness and coordination in shared workspaces', *Conference proceedings on Computer-supported cooperative work*, volume 0, pp. 107-114.

Dix, A., Finlay J., Abowd G., and Beale R. (1993): 'Human-Computer Interaction', *Prentice Hall*.

Grudin J. (1994): 'Computer-supported cooperative work: history and focus', *IEEE Comput*, pp. 19-26.

Hofte H., et al, (1995): 'CSCW Infrastructure Research at TRC', *ACM SIGOIS Bulletin*, vol. 15.

Segal L., (1995): 'Designing Team Workstations: The Choreography of Teamwork', *Local Applications of the Ecological Approach to Human-Machine Systems*, pp. 392-415.

Carl Gutwin, Greenberg S., (1998): 'Design for individuals, Design for Groups: Tradeoff between power and workspace awreness', *ACM Computer Supported Cooperative Work*, pp. 383-397.

Dix, A., Finlay J., Abowd G., and Beale R. (1998): 'Human-Computer Interaction', *Prentice Hall*.

Simone C., et al, (1999): 'Interoperability as a means of articulation work', *WACC '99: Proceedings of the international joint conference on Work activities coordination and collaboration*, pp. 39-48.

J. Fjermestad and S. R. Hiltz. (2000): 'A descriptive evaluation of group support systems case and field studies', *Journal of Management Information Systems*, pp. 115-159.

JÖrg R., Claus U. (2000): 'Developing synchronous collaborative applications with TeamComponents', *Fourth International Conference on the Design of Cooperative Systems*.

Ladu, R. and Parakash, A. (2000): 'Developing adaptive groupware applications using a mobile component framework', *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 107-116.

Riehle D., (2000): 'Framework Design: a Role Modeling Approach Dissertation', *ETH Zurich*.

Robert, D. and Briggs, R. (2000): 'GSS for presentation support', *Communications of the ACM*, pp. 43-9197.

Slagter, R., Biemans, M. and Hofte, H. (2001): 'Evolution in use of groupware: facilitating tailoring to the extreme', *Groupware, 2001. Proceedings. Seventh International Workshop*, pp. 68-73.

Carl Gutwin, Greenberg S., (2002): 'A Descriptive Framework of Workspace Awareness for Real-Time Groupware', *Journal Computer Supported Cooperative Work*, pp. 72-81.

Martínez A., et al, (2003): 'Towards an XML-based model for the representation of collaborative action', *conference on computer support for collaborative learning*, pp. 14-18.

Carl Gutwin, Reagan Penner, Kevin A. Schneider (2004): 'Group Awareness in Distributed Software Development', *CSCW*, pp. 72-81.

Carl Gutwin, et al, (2005): 'Supporting Group Awareness in Distributed Software Development', *Engineering Human Computer Interaction and Interactive System*, vol. 3425, pp. 383-397.

Akhil Mehra, et al, (2005): 'A generic approach to supporting diagram differencing and merging for collaborative design', *ASE*, pp. 204-213.

Holzinger, (2005): 'Usability engineering methods for software developers', *Communication of the ACM* vol. 48(1), pp. 71-74.

Bratitsis T., Dimitracopoulou A., (2006): 'Monitoring and analyzing group interactions in asynchronous discussion with the DIAS system', *12th international workshop on groupware*, pp. 54-61.

Frost and Sullivan (2007): 'Meetings around the world: The Impact of Collaboration on Business Performance'.

Schümmer, T. and Lukosch, S. (2007): *Patterns for Computer-Mediated Interaction*, John Wiley & Sons, Ltd.

Victor M. R. P., Maria Dolores Lozano, José A. Gallud, Ricardo T., Maria L. Rodríguez, José L. Garrido, Manuel Noguera, Maria V. Hurtado (2008): 'Extending and Supporting Featured User Interface Models for the Development of Groupware Applications', *J. UCS 14(19)*, pp. 3053-3070.

Rafael Duque, Manuel Noguera, Crescencio Bravo, José Luis Garrido and Maria Luisa Rodríguez (2009): '*Construction of interaction observation systems for collaboration analysis in groupware applications*', Advances in Engineering Software 40(12): pp. 1242-1250.

Ana B. Pelegrina, et al, (2010): 'Integrating Groupware Applications into Shared Workspaces', *RCIS*, pp. 557-568.

Briggs, R., Kolfschoten, G., Vreede, Gert-jan, Albrecht, C., and Lukosch, S. (2010). 'Facilitator in a Box: Computer Assisted Collaboration Engineering and Process Support Systems for Rapid Development of Collaborative Applications for High-Value Tasks' *Information Systems*, pp. 1-10.

Mametjanov, A., Kjeldgaard, D., Pettepier, T., Albrecht, C., Lukosch, S., and Briggs, R. (2011): 'ARCADE: Action-centered Rapid Collaborative Application Development and Execution', *Hawaii International Conference on Systems Sciences,* volume 0, pp. 1-10.

In: Nolte, A.; Prilla, M.; Lukosch, S.; Kolfschoten, G. and Herrmann, T.: Proceedings of the 1st International Workshop on Collaborative Usage and Development of Models and Visualizations at the ECSCW 2011 (CollabViz 2011)

# Practical insights into collaborative drafting of organizational processes

Selim Erol
Institute of Information Systems and New Media
Vienna University of Economics and Business
serol@*wu.ac.at*

**Abstract.** Business process modeling cannot be seen isolated from the larger context – business process design, engineering and management. We consider business process modeling and the closely related development of graphical representations of process models as a social activity by nature. In this paper we present findings from a series of cross-industry in-depth interviews of practitioners in the domain of business process design and engineering which was found to strongly support this assumption and offers new insights into the collaborative practice of process modeling. To describe the social practice of business process modeling the interview data was analyzed and interpreted using an activity-theoretic perspective. Subsequently, a generic set of recommendations was derived that can be used as a starting point to design software environments that effectively support collaboration in process modeling and (re-)design.

## Introduction

Business process modeling has become a common practice in organizations that have recognized that describing business processes in a structured way is the basis for effective business process improvement. However, business process modeling cannot be seen isolated from it's larger context – business process (re-)design, engineering and management. In this paper process modeling is understood as an activity which is inherently embedded in the context of a process (re-)design activity. Understanding the characteristics of collaboration in process modeling therefore requires to investigate the practice of process (re-)design activities in organizations. We present findings from a series of cross-industry in-depth interviews of practitioners in the domain (domain experts) which was found to support the above assumptions.

We used an activity-theoretic perspective to analyze, interpret and structure the interview data. Finally, a generic set of recommendations was derived that can be used as a starting point to design software environments to support collaborative process modeling and (re-)design activities.

# Qualitative interviews with practitioners

The practitioners (domain experts) represent a broad range regarding the industry (telecom, oil, gaming, banking, insurance, manufacturing, consulting) and role in process modeling. The practitioners were selected through the professional network of the author, through a forum of BPM experts and through a telephone survey in Austria's leading organizations. All interviews except two were audiotaped and transcribed. In sum twelve interviews were conducted throughout a three months period. The interviews were conducted using a open-ended semi-structured approach. The interview guideline contained questions to clarify the experts expertise in the field and questions that addressed the characteristics of collaboration in process (re-)design activities. As the interviews were conducted recently this summary of findings has to be seen as preliminary. However, we were able to identify main concepts prevalent throughout the interview data.

# Contextual analysis of process modeling in practice

Activity theory (AT) is an approach that has gained increasing interest in the research field of computer-supported collaboration (Engeström, 2008). It has been applied to analyze and describe various collaboration domains from an analytical and conceptual viewpoint, e.g. health care (e.g. Engeström, 1995; Bardram et al., 2011), software design (e.g. Fjeld et al., 2002; Barthelmess and Anderson, 2002; Hemetsberger, 2009), learning environments (e.g. Jonasson, 1991; Collis, 2004).

According to AT an activity is the “minimal meaningful context” to study individual human actions (Kuuttii, 1992). It is argued that in contrast to individual goal-oriented actions an activity is driven by a collective motive. It is the collective motive of an activity that makes individual actions meaningful and understandable (Engeström, 2001). Engeströms structural model of an activity system (Engeström, 1987) is based on a threefold relationship between subject, object and community. All these relationships can be mediated by three types of mediators, namely tools, rules and division of work (Kapetilinin, 1995). Additionally

Engeström describes AT in the form of five principles: (1) collective, artifact-mediated and object-oriented activity system as the prime unit of analysis, (2) multi-voicedness of activity systems, (3) historicity of activity systems, (4) contradictions as sources of change and development, (4) expansive transformations in activity (Engeström, 2000).

In the following we will suggest the activity system of business process (re-)design as the minimal meaningful context for studying collaborative process modeling (figure 1). We will discuss analyze and interpret the interview data against the this activity system.

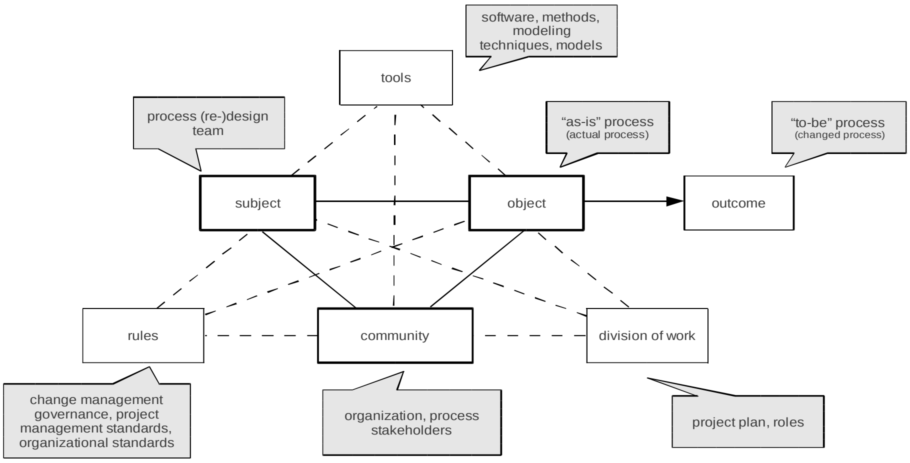


*Figure 1: Structural model of business process (re-)design activity*

*Principle 1: Collective, artifact-mediated and object-oriented activity system as the prime unit of analysis.* A key idea of AT is that an activity is a collective phenomenon emerging through goal-oriented individual actions. Specific goals are subordinate to the collective motive of the entire activity system and only can be understood against this background. In the interviews we found strong evidence that practitioners rather think in terms of process (re-)design activities or even process improvement activities than in terms of process modeling when asked about the collaborative practice in modeling. *"Enterprises do not pay for a process modeling activity rather they pay for a process improvement or a software implementation activity" (E09)*. Similarly the object and outcome of a collaborative (re-)design activity is mostly referred to as the process rather than the process model. Process models were reported to be used mainly as a mediating artifact to support communication, argumentation and validation during design rather than being the primary object of process (re-)design. This is also supported by the fact that almost all interviewees argued that they are quite indifferent about the modeling formalism to be used. Similarly interviewees almost unanimously regard the modeling software to be of minor importance though the documentation and sharing of process descriptions is regarded important. The interview data clearly reveals the importance of coordinative and communicative activities in process (re-)design.

*Principle 2: Diversity of community.* According to AT an individual subject's actions towards an object or outcome are strongly related to the community it belongs to. In the interviews conducted a general tendency is found that the community that has a stake in a process is generally large. Hence, for specific goals (e.g. elicitation of knowledge and feedback collection) small groups are formed as this is perceived more effective than involving the whole community. Large groups were reported to be only the exception to the rule and were formed only in kick--off workshops where the objectives, motivates and scope of a process (re-)design effort were presented to a larger audience. Three of the interviewees reported that such social events led to an improved awareness of colleagues involved in the same process. *".. I had projects where people participating in an identical process did not know each other, it was only through the kick-off meeting that people spread over different departments and floors got to know each other .. Naturally, it is more difficult to implement small process improvements when people do not know each other .. " (E02).* As the community directly or indirectly involved in a process (re-)design activity is large also multiple points of view, traditions and interests are existent. The analysis of interview data reveals that coordinative activities dominate over creative activities such as modeling. A continuous forth and back (review cycle, feedback loop) between stakeholders and modelers regarding the formalization of a process has been repeatedly mentioned in the interviews. To communicate results of process a variety of representations were reported to be used. Regarding the representational style of process models practice reveals that textual descriptions either unstructured or structured in the form of tables, lists and forms are equally used with graphical representations. *"The world is divided .. Our process knowledge portal supports two views. One can see a process both textual and graphical. We have run reports [on the usage of representational styles]. Which reveals a 50 to 50 distribution, who uses what. Personally I prefer diagrams, colleagues prefer tabular representations, because they can use it like a checklist. I prefer to see the big picture, they like to read textual descriptions behind the activities." (E10).*

*Principle 3: Historicity of activity system.* Activity systems carry with them a history that reflects the experiences of the individuals involved. Following AT the knowledge and experiences of a community are engraved in the artifacts it produces. In fact, several interviewees referred to historical aspects in order to explain why process (re-)design is performed in a specific way, e.g. why they use a specific methodology, modeling technique, notation or software. For example, one interviewee reported that they shifted from a centralized approach of process documentation with a single repository of process models and a single modeling technique to a decentralized approach were the main organizational units can autonomously decide how to conduct a process (re-)design effort. Another interviewee reported that he has to adjust the terminology used in process (re-)design projects as some individuals have had bad experiences with process re-engineering approaches in the past. *"Process management is fashionable today and commonly accepted. But until a year ago some people did not even want to hear the word 'process' as this was associated with consultants drawing some odd process charts .. " (E07).* Several interviewees give evidence that maintaining a revision history of process models is not valued as a source of knowledge for process (re-)design. Rather, process documentation is maintained in accustomed document management systems.

*Principle 4: Contradictions as sources of change and development.* Contradictions result from incompatibilities between the elements of an activity system. Contradictions are the driver for situational adaption of an activity system. For example, a modeling tool may not fulfill the requirements of a process (re-)design activity as notational elements to model organizational units are missing. Also conflicts may arise between stakeholders regarding the granularity (details to include) in the model. Contradictions emerge as well when stakeholders have to come to an agreement regarding a newly designed process. However, in the interview data we found evidence that process design takes place in an highly iterative manner between stakeholders and modelers. Thus, interviewees did not mention severe conflicts during process (re-)design to be an issue. Another example mentioned by interviewees is the gap between the stakeholders required and the stakeholders having capacity to participate in a process (re-)design effort. All these contradictions may influence the course a collaborative (re-)design activity takes, whether models are accepted and reused by a community.

*Principle 5: Expansive transformations in activity.* As contradictions may become aggravated over lengthy periods of time individuals begin to question established artifacts, norms, rules and procedures. Therefore an activity is evolving into a new activity system. For example, in two cases it was reported that rigid implementation of process governance standards failed due to the resistance of departments which did not follow the standards due to reasons of inadequacy and fear of transparency. This led to a more flexible and decentralized approach where departments were able to adapt corporate conventions to their needs or to use their own conventions and tools. Other practitioners pointed to the fact that they have gradually adapted the software tools used for process modeling and maintenance as tools did not meet specific requirements. The same is experienced with project methodologies or workflow procedures determining the way a community collaborates in a re-design activity.

# Conclusion and Outlook

In the preceding section we have used Activity Theory (AT) to identify and discuss the minimal meaningful context of process modeling – business process (re-)design. Though, only selected issues have been outlined in this paper we found that for understanding collaboration in process modeling especially the non-expert/expert interaction, diversity of the community and the developmental character of process (re-)design a has to be investigated in more depth. In future research activities we will use these findings to derive general guidelines for designing respective software environments. In table 1 a set of six recommendations is suggested which is not meant to be complete but can be seen as complementary to other work in the field (e.g. Renger & Kolfschoten, 2008; de Vreede, 2009; Herrmann & Nolte, 2010-2011, Rosemann, 2008; Rittgen, 2009; Erol et al., 2010).

| | |
|---|---|
| **R1:** | Integrate the larger context of process modeling. E.g. a process improvement, change management, requirements elicitation, system development, .. (← P1) |
| **R2:** | Support the shift from close (face-to-face, synchronous, co-located) to loosely coupled (asynchronous, distributed) collaboration in process (re-)design (← P2) |
| **R3:** | Provide means to use diverse representation styles, notations and tools for describing a process for a diverse community of stakeholders (← P2) |
| **R3:** | Provide mechanisms that allow the interaction with process models for a broad community and at the same time ensure the stability of process models (← P2) |
| **R3:** | Support the shift from initial process model creation activities to long-term process model maintenance (← P1, P3) |
| **R4:** | Support the smooth adaption of process modeling techniques and tools to situational needs (← P4, P5) |
| **R6:** | Consider the twofold nature of process models being primarily a mediating artifact for the design activity and the object of modeling (← P1, P4) |

Table 1: recommendations for designing collaborative process modeling environments (references in brackets refer to the principles of AT)

# References

Bardram, J. and Doryab, A. (2011): Activity analysis: applying activity theory to analyze complex work in hospitals. Proceedings of the ACM 2011 conference on Computer supported cooperative work, pp. 455–464.

Barthelmess, P. and K.M. Anderson (2002): A View of Software Development Environments Based on Activity Theory. Computer Supperted Cooperative Work, vol. 11, nos. 1–2, pp. 13–37.

Collis, B. and Margaryan, A. (2004): Applying activity theory to computer-supported collaborative learning and work-based activities in corporate settings. Educational Technology Research and Development, vol. 52, no. 4, pp. 38-52, Springer.

Engeström, Y. (1995): Objects, contradictions and collaboration in medical cognition: an activity-theoretical perspective, Journal of Artificial Intelligence in Medicine, Elsevier, vol. 7, no. 5, pp. 395—412

Engeström, Y. (1987): Learning by expanding. Retrieved June 14, 2011, from http://communication.ucsd.edu/MCA/Paper/Engestrom/expanding/toc.htm

Engeström, Y. (2000): Activity theory as a framework for analyzing and redesigning work, Journal of Ergonomics, Elsevier, vol. 43, no. 7, pp. 960-974

Engeström, Y. (2001): Expansive learning at work: Toward an activity theoretical reconceptualization, Routledge Journal of Education and Work, vol.14, pp. 133–156

Engeström, Y. (2008): The future of activity theory: a rough draft. Keynote lecture presented at the ISCAR Conference in San Diego, Sept. 8-13, 2008

de Vreede (2009): How Interactive Whiteboards Can be Used to Support Collaborative Modeling. Journal of Universal Computer Science. vol. 15, no. 16.

Fjeld, M. and Lauche, K. and Bichsel, M. and Voorhorst, F. and Krueger, H. and Rauterberg, M. (2002): Physical and virtual tools: Activity theory applied to the design of groupware. Journal of CSCW, vol. 11, no. 1, pp. 153-180, Springer.

Herrmann, T. and Hoffmann, M. and Loser, K.U., Moysich, K. (2005): Semistructured models are surprisingly useful for user-centered design. In: Dieng, R.; Giboin, A., Karsenty, L., De Michelis, G. (Eds.): Designing cooperative systems. Proceedings of COOP2000. Amsterdam: IOC press. pp. 159-174

Herrmann, T. and Nolte, A. (2010): he integration of collaborative process modeling and electronic brainstorming in co-located meetings. Journal of Collaboration and Technology, Springer.

Hemetsberger, A. and Reinhardt,C.: Collective Development in Open-Source Communities: An Activity Theoretical Perspective on Successful Online Collaboration. Journal of Organization Studies 30(09): 987–1008, SAGE.

Kettinger, W. and S. Guha (1997): 'Business process change: a study of methodologies, techniques, and tools'. MIS Quarterly, vol. 21, no. 1, pp. 55-80.

Kuutti, K. (1991): The concept of Activity as a Basic Unit of Analysis for CSCW Research. Proceedings of the Second European Conference on CSCW. Amsterdam.

Jonasson, D.H. and Rohrer-Murphy, L. (1999): Educational Technology, Research and Development; 1999; 47, 1; Research Library, pg. 61, Springer.

Nardi, B. (1996b): Activity Theory and Human-Computer Interaction. In Nardi, B. (ed.) (1996): Context and Consciousness, MIT Press, pp. 12-13.

Erol, S. and Mödritscher, F. and Neumann, G. (2010): A Meta-Design Approach To Collaborative Process Modeling. Presented at DIS 2010 Conference / Workshop on Open Design Spaces in Aarhus, Denmark, August 2010. Published in International Reports on Socio-Informatics.

Renger, M., G. Kolfschoten, and G. Vreede (2008): Challenges in Collaborative Modeling: A Literature Review, Vol. 10 of LNBIP, pp. 61–77. Springer.

Rittgen, P. (2009): Collaborative modeling-a design science approach. HICSS'09. 42nd Hawaii International Conference on System Sciences, 2009.

Rosemann, M. (2008): Understanding and Impacting the Practice of Business Process Management. In M. Dumas, M. Reichert, and M. Shan, editors, BPM 2008, pp. 2, Springer.

In: Nolte, A.; Prilla, M.; Lukosch, S.; Kolfschoten, G. and Herrmann, T.: Proceedings of the 1st International Workshop on Collaborative Usage and Development of Models and Visualizations at the ECSCW 2011 (CollabViz 2011)

# Process models: Neutral ground for collaboration, but power matters

Alexander Nolte[1], Michael Prilla[1]
[1]Information and Technology Management, University of Bochum, Germany
nolte@iaw.rub.de, prilla@iaw.rub.de

**Abstract.** Models and other process visualizations are common artifacts in organizations to visualize, analyze and sustain processes. They also serve as artifacts for communication. In these settings, models serve as neutral ground taking away anxieties usually arising when different parties work together. Models can also become tools of power enabling inferior participants to state their opinion or becoming tools superior participants want to control. Facilitation of model usage and development can give room to the positive aspects of this usage and diminish possible downsides. This paper deals with the question whether these effects can also be achieved in situations in which people use models on their own. As we found in a study, some of these effects are present without facilitation, but there is some work remaining to support all of them in practice.

## Introduction

Visualizations of work such as process models are established tools in modern organizations. They support people in making perspectives explicit, understanding the work of others, jointly planning work and communicating about it (cf. Suchman 1995, Herrmann et al. 2004b, Prilla 2010). This is mirrored by many methods using models and other visualizations for the design of cooperation support (e.g. Beyer and Holtzblatt 1998, Conklin 2005, Herrmann 2009). Most of these methods rely on expert facilitators: Users do not use or manipulate visualizations directly, but their utterances are connected to visualizations by experts during or after the interaction. Thus, the usage of models by non-experts depends on the availability of experts. Besides such settings, models are rarely used by other people (cf. Wand and Weber 2002, Prilla 2010). This slows down model development and prevents positive effects of models on cooperation.

People are capable of using models to support communication and manipulating them if they are given adequate means to do so (cf. Herrmann 2009, Prilla 2010, Prilla and Nolte 2010). Thus, adequate support of **self-directed interaction with models** (interaction without facilitation during model usage or manipulation) can diminish the problems of expert-driven model interaction and preserve the benefits of it. Thus, we created a prototype for such interaction and an experimental setting to explore users' interaction with process models. Through this we wanted to explore whether the benefits of models in expert-facilitated settings can also be reached in self-directed settings. In this paper, we report on results from this approach.

In what follows, we describe potentials and problems of model interaction. After that, we describe our experimental setting and the results stemming from our experiments. We then discuss our findings and elaborate on further work to be done for the implementation of self-directed interaction with models.

# Potentials and pitfalls of collaborative model usage

There are several contributions from CSCW and related disciplines providing insights into potentials and pitfalls of the model usage we intend to support. Among others, we identified the following insights to be most influential for this:

- **Models for the exchange of perspectives and negotiation in grounding:** Models can be boundary objects (Star 1989), making perspectives explicit and support people in exchanging these perspectives and in negotiating common understanding (cf. Davies et al. 2004, Herrmann and Hoffmann 2005).
- **Models support communication:** Visualizations can make work visible to others (Suchman 1995), help designers from different backgrounds to find a common solution (Herrmann et al. 2004a), support communication about past activities and trigger communication (Fleck and Fitzpatrick 2006).
- **Models equalize politics and hierarchies:** Working with models can equalize differences in opinions and hierarchies among cooperators (Samarasan 1988, Herrmann et al. 2004b). However, in practice this work includes both the "artful crafting of peoples' stories" and political or hierarchical influences leading to "strategic manipulation of images" (Suchman 1995). Facilitation of group modeling can diminish unwanted influences (Samarasan 1988, Herrmann 2009).

The advantages described above stem from facilitated model usage. Therefore, we cannot take these benefits for granted in self-directed model interaction. Also, downsides such as unwanted influence may reoccur if we reduce the influence of facilitators and let people use models on their own. Dealing with that needs exploring model interaction and analyzing it properly:

- Concerning its **applicability for negotiation processes**, we need to analyze model-related negotiation processes during self-directed model usage. For this, Beers et al. (2005) name primitives of negotiation such as contributing own perspectives, verifying the understanding of other perspectives, clarifying contributions and accepting or rejecting it.
- For the analysis of **communication about models** we need to look for model references in communication. Typical elements for this can be pointing to a model or referring to parts of a model during communication.
- In order to explore whether self-directed model usage has an **effect on political and hierarchical influences** on interaction, we need to analyze the conversations between actors using models according to arguments exchanged, decisions made and rationales behind them.

# Setting: A prototype and environment for non-expert model interaction

The exploratory study was conducted with a prototype built based on experiences from prior work (c.f. Herrmann et al., 2010), which enables users to contribute to a process model without the need to be familiar with the respective process modeling language. It uses the SeeMe modeling language, which has been shown to be easily understood even by inexperienced people (cf. Herrmann et al. 2004a, Herrmann 2009). This prototype is coupled with an environment providing a large rear projection touch screen used to visualize process models and users' contributions to them as well as to manipulate resulting models via touch interaction (see Figure 1 for a glimpse of the environment). This environment provides an easy to use and intuitive interaction with models and is thus ideal for

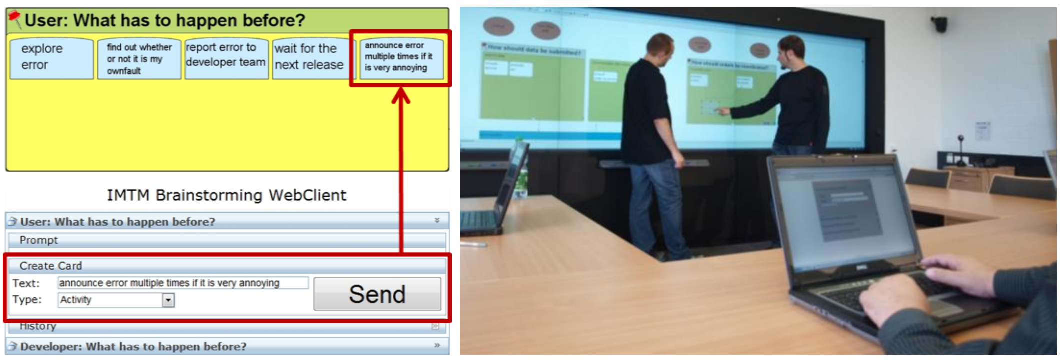


Figure 1: Contributing to a model from a web interface and transformation to a labeled model element (left) and self-directed interaction with process models on a large touch screen (right).

our purpose of exploring self-directed model interaction.

In our experiments, pairs of participants interacted with process models. We used scenarios of processes they were familiar with, which included two different roles (see Table 1). Each role was taken by one participant. We conducted five experiments (three covered scenario one) with two participants each, lasting about 30-45 minutes. We included different kinds of self-directed model usage into the experiments. First, participants were asked to add necessary parts of the process from the scenario to their own process model. After that, they had to explain the resulting models to each other and identify differences concerning both content and sequence of actions. After that, they were asked to articulate differences and similarities they found. During the experiment a facilitator guided the participants through the script of actions, but did not intervene in any model-related tasks.

The participants we worked with differed in terms of hierarchies between them (see Table 1). For two pairs, one participant was ranked significantly higher than the other and for the other three pairs, there was no big gap in hierarchies.

Table 1: Participants of the experiments and hierarchies between them.

| **Pair** | **Scenario** | **Participant 1** | **Participant 2** | **Hierarchy** |
|---|---|---|---|---|
| P1 | (1) Bug processing in software dev. | Project manager | Junior Developer | Yes |
| P2 | (2) Book ordering in a library | Library owner | Library clerk | Yes |
| P3 | (1) Bug processing in software dev. | Software user | Software developer | No |
| P4 | (1) Bug processing in software dev. | Software user | Senior developer | No |
| P5 | (2) Book ordering in a library | Library user | Library clerk | No |

For analysis, we videotaped the workshops and an observer made notes. Afterwards, we analyzed this material according to the criteria described above.

# Insights into self-directed model interaction

We observed models to support and influence the communication of participants in many ways. They oftentimes served as artifacts of common ground and reference. Unfortunately, we also observed influences of hierarchies. This shows that models can be used for grounding, but that power still matters in their usage. In what follows, we describe a selection of the most remarkable findings.

**Models as means for the creation of neutral ground:** In the experiments, the model-related tasks conducted by the participants fostered the creation of neutral ground. For example, we observed that visualizing the perspective of participants and communication about them fostered the understanding for each other’s work. By e.g. pointing to models during discussion, the participants were able to identify differences and to cope with them on neutral ground and without the help of a facilitator. In addition, participants told us that the preparation of models during the contribution of activities to their own model helped them *"... to create a compressed visualization of the own view..."* which made *"...the following discussion much easier..."* (developer from P3).

**Models as a result of negotiation:** During the discussion and – in absence of hierarchies – during the negotiation of differences, models proved to equalize gaps in different opinions. For example, the user of pair P4 (c.f. Table 1) criticized a lack of awareness on *"...the current state of a bug and the current priorities of development..."*. In contrast to that, the developer stated that he would *"...avoid giving feedback or even talking to the users..."* as this would distract him and slow him down, causing the bug to last longer. This discussion was triggered by the fact that during the comparison they had found that the user had included a *feedback*-activity into the process of bug processing whereas the developer had not. After a short discussion they agreed to a solution: The user would receive better feedback on bug processing while exact details would be left to the developer. This example shows how communication can be triggered in self-directed model interaction and how it can support the negotiation processes.

**Models as a result of hierarchical decision:** In contrast to the description above we experienced that hierarchy plays a decisive role in negotiation processes related to models. This was especially present in pairs P1 and P2, who had a huge difference in status. For P1, this resulted in the developer oftentimes instantly adopting the view of the user without any notable negotiation. When it came to a discussion about what is considered to be a bug, the user stated that *"...anything that does not work as expected is a bug..."* while the developer first considered a bug to be *"...a malfunction compared to how it is implemented..."*. However, after the user had explained his notion, the developer inclined to this view without any discussion possibly although he felt he was right. This shows that self-directed model interaction cannot prevent hierarchies from being an influence.

Summing up, we found all aspects discussed above in the observed interaction: self-directed work with models triggered communication and models were used as a reference in communication. Moreover, we found the benefit of models for perspective exchange and negotiation of common ground as well. For unwanted influences such as hierarchical decisions, we need to find solutions in order to consider self-directed model interaction to be an alternative to facilitated settings.

# Conclusion and future work

In this paper, we report on an approach in enabling people to work with models on their own, preserving positive aspects of models for collaboration and diminishing possible problems. Results from our experiment indicate that – up to a certain extent – perspective exchange and negotiation about processes does not require content related facilitation and can be done self-directed. Given the right means, users can express their perspectives on their own and are able to discuss and negotiate them. Furthermore, perspective exchange and discussion was not decisively influenced by hierarchies. However, when manipulation of process parts requires negotiation, hierarchy influences the outcome.

In the future we will conduct further experiments to gain more sustainable data on the insights described before – especially dealing with hierarchies will be part of this work. Currently, there are a lot of questions remaining for our work:

- How to compose models from different perspectives and negotiate them with special regard to hierarchy and how does group composition affect this?
- How is self-directed model usage and negotiation affected by the separation or intertwining of discussion and design with phases of assessment?
- To what extent are non-expert modelers capable of dealing with formalism and how can the functionality of a tool support them adequately?
- How much support can software provide for self-directed model interaction and when are modeling experts and facilitators required?

# References

Beers, P. J., H. P. A. Boshuizen, P. A. Kirschner and W. H. Gijselaers (2005). Computer support for knowledge construction in collaborative learning environments. Computers in Human Behavior, vol. 21, pp. 623–644.

Beyer, H. and K. Holtzblatt (1998). Contextual design: defining customer-centered systems. Morgan Kaufmann.

Conklin, J. (2005). Dialogue Mapping. Wiley, Chichester.

Davies, I., P. Green, M. Rosemann, S. Gallo, A. Brisbane and A. Ipswich (2004). Conceptual Modelling - What and Why in Current Practice: Conceptual modeling: ER 2004: 23rd International Conference on Conceptual Modeling. New York: Springer, pp. 668–681.

Fleck, R. and G. Fitzpatrick (2006). Supporting collaborative reflection with passive image capture. In Hassanaly, P.; Herrmann, T.; Kunau, G.; Zacklad, M. (eds): Cooperative Systems Design. Seamless Integration of Artifacts and Conversations - Enhanced Concepts of Infrastructure for CommunicationProceedings of COOP 2006. IOS Press.

Herrmann, T., M. Hoffmann, G. Kunau and K. U. Loser (2004). A modelling method for the development of groupware applications as socio-technical systems. Behaviour & Information Technology, vol. 23. no. 2, pp. 119-135.

Herrmann, Thomas (2009). Systems Design with the Socio-Technical Walkthrough. In Whitworth, Brian; Moor, Aldo de (eds): Handbook of Research on Socio-Technical Design and Social Networking Systems. Information Science Reference.

Herrmann, Thomas and Marcel Hoffmann (2005). The Metamorphoses of Workflow Projects in their Early Stages. Computer Supported Cooperative Work, vol. 14. no. 5, pp. 399–432.

Herrmann, Thomas, Gabriele Kunau, Kai-Uwe Loser and Natalja Menold (2004b). Sociotechnical Walkthrough: Designing Technology along Work Processes. In Clement, Andrew; Cindio, Fiorella; Oostveen, Anne-Marie; Schuler, Doug; van den Besselaar, Peter (eds): Artful Integration: Interweaving Media, Materials and Practices Proceedings of the eighth Participatory Design Conference 2004. ACM Press, pp. 132–141.

Herrmann, Thomas, Alexander Nolte and Michael Prilla (2011). The Integration of Awareness, Creativity Support and Collaborative Process Modeling in Colocated Meetings. International Journal on Computer-Supported Cooperative Work (IJCSCW). Special Issue on Awareness (accepted).

Prilla, Michael (2010). Models, Social Tagging and Knowledge Management – A fruitful Combination for Process Improvement. In Rinderle-Ma, Stefanie; Sadiq, Shazia; Leymann, Frank (eds): BPM International Workshops 2009.

Prilla, Michael and Alexander Nolte (2010). Fostering self-direction in participatory process design: Proceedings of the Participatory Design Conference 2010 (PDC 2010).

Samarasan, D. (1988). Collaborative modeling and negotiation. Conference on Supporting Group Work, pp. 9-21.

Star, Susan Leigh (1989). The structure of ill-structured solutions: boundary objects and heterogeneous distributed problem solving. In Gasser, Less; Huhns, Michael H. (eds): Distributed Artificial Intelligence. Morgan Kaufmann Publishers Inc., pp. 37-54.

Suchman, L. (1995). Making work visible. Communications of the ACM, vol. 38. no. 9.

Wand, Y. and R. Weber (2002). Research Commentary: Information Systems and Conceptual Modeling. A Research Agenda. Information Systems Research, vol. 13. no. 4, pp. 363–376.