5-2009

Collaborative Modeling Architecture: Lessons Learned from Deploying a Prototype

Peter Rittgen
Vlerick Leuven Gent Management School, peter.rittgen@vlerick.be

21. COLLABORATIVE MODELING ARCHITECTURE: LESSONS LEARNED FROM DEPLOYING A PROTOTYPE

Peter Rittgen

Vlerick Leuven Gent Management School, Belgium & University of Borås, Sweden
peter.rittgen@vlerick.be

Abstract

Earlier we have developed an architecture for collaborative modeling based on observations of participants in a modeling team. This architecture has been implemented in a prototype tool that supports group modeling in UML. We have deployed this tool both in real life and in a more controlled student setting. From the former we have identified problem types that can be solved with the help of architecture and tool, and from the latter we have learned how to improve the usefulness and usability of the tool itself.

1. Introduction

The collaborative nature of conceptual modeling as an instrument for describing and changing the social world has been widely accepted [1-5]. Despite this overall agreement there is little research on detailed and specific architectures for organizing the process of collaborative modeling and on tools that can support it.

In [1], for example, modeling is viewed as an information gathering dialogue where knowledge is elicited from domain experts. We adopt an extended view where modeling is rather a social and communicative process where much information is created by and through the process rather than only gathered from domain experts. We have therefore in an earlier study looked at a situation where the participants had no a priori roles and started from a similar position of having very little domain knowledge and collectively tried to make sense of a case.

[2] emphasizes the importance of natural language as the primary medium and identifies the concretization of an informal model and the abstraction of a formal model as the principle activities. We take up this distinction and refine it in section 4.1. [3] makes a similar distinction between an elicitation and a formalization dialogue and develops a modeling procedure by generalizing existing procedures for specific modeling languages.

[4] acknowledges that modeling is not only a knowledge elicitation process but also a knowledge creation and dissemination process. It is viewed as a structured conversation where sub-conversations are associated with goals and strategies (the latter are elaborated in [1]). We fully agree with that but claim that it is a specific type of conversation, namely a negotiation. This idea is implicitly present in [4] where the dialogue structure contains negotiation elements such as propose and accept. We have elaborated this point in an earlier study arriving at a more detailed negotiation model. [4] also advocates the use of controlled language and validation. We consider the latter as problematic as it has often been observed that domain experts falsely agree with a model not being fully aware of all its implications.

[5] studies the influence of situational factors on modeling (in particular, enterprise modeling). The author’s aim is to create an environment that facilitates and supports participative modeling. This work is complementary to our work.

In the following we first present the architecture that was developed earlier (section 2). We then go on to describe the first version of the prototype of a modeling tool to support this architecture (section 3). In section 4 we report on our experiences from the deployment of this prototype in two different settings. The first (section 4.1) was performed in the context of a real-life modeling project in a large manufacturing company for mobile network equipment. Here the focus was on identifying important modeling activities (or objectives) to extend the existing architecture. The second one (section 4.2) was carried out as a student experiment with a focus on the usefulness and usability of the tool.

2. Collaborative Modeling Architecture (COMA)

COMA was originally developed on the basis of a large student experiment over 3 years that investigated the modeling behavior during group modeling sessions. Data was collected with the help of the think-aloud process-tracing methodology [6, 7], the transcription of conversations during modeling and the produced models themselves. The pre-categorization was done based on the semiotic ladder [8] from organizational semiotics and generic activities were developed for the syntactic, semantic, pragmatic and social levels. Here we abstract from the syntactic and semantic levels as tools for these levels already exist in abundance.

2.1 Results on the Social Level

The social norms within a modeling team are mainly made up of rules for determining whether a proposal is accepted or rejected. We observed that these rules do not have to be logical complements which allows for situations where a proposal can be neither rejected nor accepted but requires further convincing to decide one way or the other. A termination rule was applied occasionally to force a decision if a negotiation got stuck, i.e., when there were no more changes in the individuals' convictions over an extended period of time. We witnessed two types of rules:

- *Rules of majority*, where a certain number of group members had to support or oppose a proposal in order for the whole group to accept or reject it (e.g., more than half). A tie-break rule was sometimes specified (e.g., for the case of an equal number of supporters and opponents). The tie-break could involve seniority issues.
- *Rules of seniority*, where the weight of a group member's support or opposition was related to his or her status within the group. This status could be acquired (e.g., by experience) or associated with a position to which the member was appointed. A frequent example of this was the case of a more experienced modeler who was considered as the leader by the group and took decisions on their behalf. The other members filled the role of consultants in such a case.

These rules were sometimes set up explicitly before the group began their work, or in an early phase of this work. But in most cases they rather emerged as the result of each member's behavior. Individuals making regular contributions of high quality were likely to acquire seniority. In homogeneous teams majority rules were used more often.

2.2 Results on the Pragmatic Level

The majority of the activities on the pragmatic level were associated with negotiation. This is surprising as modeling is often pictured as an intuitive act that is mostly the product of one modeler (e.g., a consultant) that possibly receives some input from other stakeholders in the modeling process (e.g., domain experts from the respective departments). An analysis of the dependencies between activities on the pragmatic level revealed a workflow-like structure that follows a certain pattern. This pattern is shown in Fig. 1.

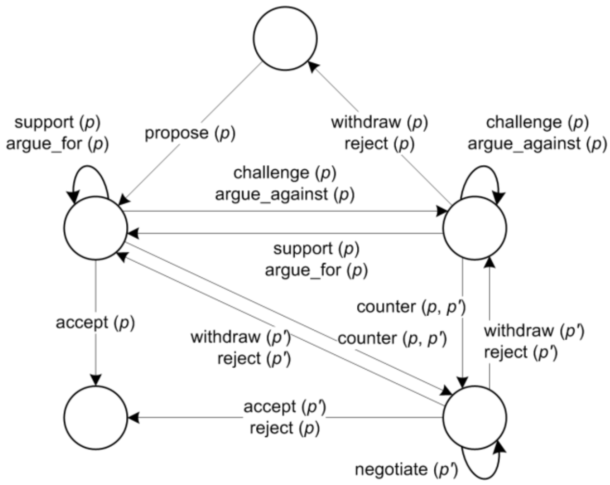

Fig. 1. Negotiation pattern

It consists of an initial and reject state at the top, a state where acceptance is favored (upper left-hand corner), a state where rejection is favored (upper right-hand corner), a recursive sub-state for negotiating a counter-proposal (lower right-hand corner) and an accept state (lower left-hand corner). Each of the states allows for a set of certain pragmatic activities that take the negotiation to a different state. We have left out the parameters concerning the modeler who performs the activity and the argument (if present). In general any modeler can perform any activity but there are a few rules to be observed. A modeler making a proposal is implicitly assumed to support it. He is the only one who may withdraw it. A counter-argument is brought up by a different modeler but a counter-proposal can also be made by the proponent of the original proposal, e.g., to accommodate counter-arguments. With the help of the pattern of Figure 3 we can control the negotiation component of a modeling support system. On the other levels we were not able to discover an equally strong pattern of activities. This will affect the kind of support a tool can provide at the language level.

3 COMA Tool

The architecture of a collaborative modeling tool, i.e., a system that supports a group in developing models, is still under investigation. Some authors have suggested groupware systems that help teams in collective sense-making [3, 9-11] which is an important part of the modeling process. [11] reports on an approach, Compendium, that is the result of 15 years of experience. Compendium combines three different areas: meeting facilitation, graphical hypertext and conceptual frameworks. To make them work, facilitation is viewed as essential to remove the cognitive overhead for the group members, i.e., the necessity to develop hypertext literacy, which cannot be assumed in all participants.

On the technology side, the critical elements are question-based templates, metadata and maps. They allow participants to move freely between different levels of abstraction and formalization as the need dictates. The question-based templates guide the process by supplying relevant questions, the answers to which will lead the group towards a better understanding of the problem and towards the development of appropriate solutions (e.g., models). The metadata is used to provide additional information that is also considered relevant but was not anticipated in the templates or lies at the intersection of templates. The maps have a hierarchical structure and the same concept can appear in different maps so that its use in different contexts can be understood. This feature is called transclusion.

Groupware systems for collective sense-making, as the one mentioned, address an important issue in collaborative modeling. They can therefore be used as the core of a collaborative modeling tool. So far these systems are typically tailored for specific modeling languages though (in the case of Compendium, World Modeling Framework and Issue-Based Information System). For a collaborative modeling tool they need to be more modular so that any modeling language can be "plugged in" (e.g., other enterprise or information systems modeling languages). In addition, there is also the need for a negotiation component that facilitates structured arguments and decisions regarding modeling choices. The model shown in Figure 3 can function as a workflow template controlling such a negotiation component.

The COMA architecture assumes the existence of a modeling tool that takes care of the syntactic level of the modeling process (i.e., a conventional single-user tool). Such tools exist in abundance for most modeling languages. Some of the tools even provide support on the semantic layer, e.g. by determining run-time properties of a business process such as liveness and freedom of deadlocks.
COMA is therefore primarily situated on the pragmatic and social levels. On the former it specifies distributed model negotiation, which can be seen as a special case of a distributed decision support system [12]. This is described in section 3.1.

On the social level it provides facilities to express the social norms that the group has adopted and that control the way in which negotiation takes place, in particular under which conditions a proposed change to the model is accepted. This is described in section 3.2.

3.1 Negotiation in the COMA tool

Distributed model negotiation means the coordination of the efforts of a number of modelers. The results from section 4.2 suggest that such a system must provide the following functions:

- Propose, Withdraw, Counter

- Support, Argue_for
- Challenge, Argue_against

To simplify the architecture we have decided to arrange the negotiation activities in three groups of related functions concerning proposals, supports and challenges. This implies that we drop the distinction of only two competing proposals (proposal and counter-proposal) in favor of considering any number of competing proposal (one from each proponent / modeler). It also means that we view withdrawals as less important as they can be replaced by making a new proposal (if necessary, an empty one). We have also dropped the distinction between *Support* and *Argue_for* because the former is the same as the latter without supplying the argument. The same holds for *Challenge* and *Argue_against*. The challenger is required to provide an argument (i.e., an indication of how the proposal can be improved), the supporter not.

In addition to the functions the negotiation component needs to provide each modeler with a clear overview of: The model building process and the current status of the negotiation. A view of the model building process involves:

- The current stable version of the model as agreed upon so far
- A version that contains the local changes made by the respective group member (the local model editor), and
- The proposals made by the other modelers suggesting changes to the current version.

Regarding the status of the negotiation process the modeler needs access to the following information:

- What are the arguments for and against a proposal?
- Who is in support of or against a proposal?
- What is the final decision regarding the proposal?

The model building view is divided into three areas. The upper one shows the current version that has emerged from the negotiation process so far. It is used as a reference for all other temporary versions such as the proposals and the local version. This means that suggested changes are always expressed in relation to the current version.

The lower left area contains the local version, i.e. it serves as the model editor for the group member running this particular copy of the modeling support system. It offers the model creation facilities pertaining to the syntactic level such as introducing and connecting nodes. In addition to that it provides the pragmatic functions related to making proposals. Making a proposal implies that the local version is published, i.e. made accessible to the other modelers.

The negotiation view shows a list of pros and cons for the proposal (called supports and challenges). These are the arguments supplied when the *argue_for* or *argue_against* function is performed on a proposal by a modeler. Depending on the currently active rule on the social level this determines overall acceptance or rejection of the proposal, e.g. when a required majority has been reached. A facilitator might also force a decision if the required condition cannot be reached.

An accepted proposal makes all competing proposals obsolete so they will be deleted. Observe that the deleted proposals still exist as local copies so that the respective modelers can decide to post them again as proposals if applicable. If a proposal is accepted it becomes the new current version, i.e. the upper area is updated for all group members.

The architecture that we have described so far is a direct consequence of the results from the empirical study detailed in the previous sections. The architecture supports the activities on the social, pragmatic and syntactic levels. We have built a prototype that implements this architecture and that allows us to gain further insight into the modeling process. This prototype can be employed in a number of different ways:

- It can be used to test the suggested architecture and thereby indirectly confirm the study results in a broader empirical study.
- The additional tool support in this study makes it easier to perform such a study on a larger scale, e.g. with distributed team members, and
- It also provides additional information about the modeling process that was not available in the original study.

The existence of version histories, for example, makes it possible to analyze the modeling process in a more detailed manner regarding the development stages of a model. Another example is the negotiation log that gives us a deep insight into the arguing process and the competition between different model alternatives. A study supported by this tool can therefore also contribute to the development of new theories of the modeling process.
On the practical side, the prototype can also help in detecting shortcomings and suggesting improvements. These suggestions can be related to the implementation (i.e., the tool itself) or the architecture behind it (as

outlined above). Issues such as the design of the user interface and migration to other modeling languages are important considerations here. Section 8 describes the COMA tool.

3.2 Social Norms in the COMA tool

COMA distinguishes between two rules to decide on the acceptance of proposals: Rules of majority and rules of seniority. When a rule of majority is applied, the modeling team operates in a democratic (unfacilitated) mode where each modeler has a vote of usually the same weight (different weights might be considered, though). Whether a proposal is accepted only depends on how many supporters and challengers it has. The rule specifies the minimum number of supports required, and the maximum number of challenges allowed for a proposal to be accepted. Prior to the modeling session the team has to agree on suitable parameters. The required number of supports should be at least two to avoid that a single modeler (e.g. the proponent) can make the decision alone. A maximum number of challenges of 0 would force a unanimous decision.

When a rule of seniority is applied, the modeling team has a facilitator that makes the decision. Other group members cannot directly influence the decision, but they can do so indirectly by making suitable comments (i.e., supports and challenges). The facilitator can (and indeed should) consider the supports and challenges in his or her decision. The facilitator is either determined externally or by the group (e.g., the most experienced modeler). In principle two or more facilitators might be appointed for larger groups but they have to coordinate their decisions in order to avoid confusion.

4 Deploying the COMA Tool

The COMA tool has been deployed in two different empirical studies. In the first we tried version 1 of the tool in a case study with a large industrial company. The aim of that study was to find out which kinds of problems occur in collaborative modeling and which of those can be supported by the current architecture and tool. We were also interested in finding possible extensions and improvements of the existing tool and the underlying architecture.

The second study was a student experiment that was carried out in a final-year course on Information Systems and Business Processes. Here the focus was primarily on the tool itself. We wanted to know in how far the tool supported the students in their assignments regarding business process modeling of a real-life hospital case that was somewhat adapted for educational purposes. We were also interested in learning about the usefulness and usability of the tool, i.e. changes in the functionality and the user interface required for making the tool more useful and easier to use.

4.1 A Case Study

The case study involved collaborative modeling in UML, primarily use case and activity diagrams with the ultimate purpose of requirements elicitation. The study was performed during the summer 2007 in a large company manufacturing mobile network components (called MobCom henceforth). The overall project aimed at the introduction of a new warehouse management system to replace the existing scattered landscape of "island solutions". The objective of the first phase of that project was to derive the requirements for such a system, and as part of that work an analysis of the business processes of the logistics unit (customer distribution centre). As MobCom works with the Unified Modeling Language (UML) internally, the analysis models were supposed to be in UML, too. We therefore modeled the logistics processes with the help of activity diagrams and used use cases for the requirements specification.

Fig. 2 contains a screenshot of version 1 of the COMA tool. It shows the point in time where the group is about to prepare the second version of the activity diagram for the treatment of problem goods (i.e., goods with an unclear recipient). The upper window contains the old version V001 as a point of reference. Here the participants had only collected all relevant activities. A proposal from Peter (lower right window) suggests ordering these activities in a certain sequence but Jenny (the user from which the screenshot in Fig. 2 was taken) refines it based on her more detailed knowledge of the process (lower left window). In the next step Jenny's proposal was accepted by the group and became version V002 (not visible in the screenshot).

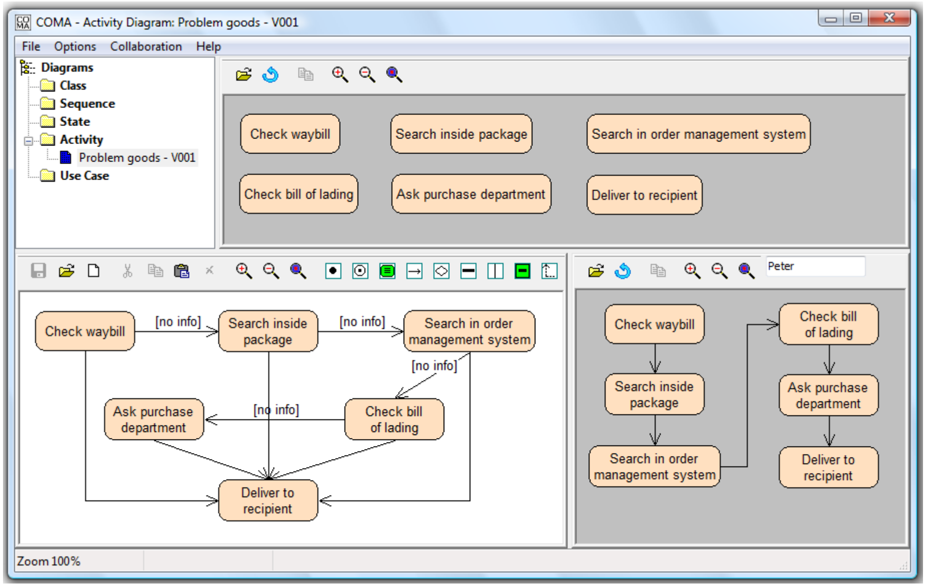

Fig. 2. Version 1 of the COMA tool

To identify the problems in collaborative modeling in a systematic way we conducted interviews with all participants. The focus was on open questions regarding the types of problems that were encountered in collaborative modeling; this was complemented by closed questions on the support that the COMA tool provided in solving these problems. We used a simple three-point Likert scale here that comprised the values good support, some support and no support.

The results from the open questions were classified along two dimensions, the problem solving domain and the degree of collaboration. The problem solving domain is the natural language domain or the modeling language domain and the associated modeling activities are usually called the concretization (of an informal model) and the abstraction (of a formal model) [2]. The degree of collaboration is a spectrum that ranges from no coupling (individual work) via loose coupling (coordination) to tight coupling (cooperation). Table 1 summarizes the results.

According to the results from the closed questions about the degree of support that the tool provides regarding the generic group modeling activities, we found that there is sufficient support for individual activities and abstraction activities. No or little help is offered by COMA when it comes to social activities in the natural language domain (see table items in italics). This suggests that the COMA tool has to be extended by functionality that offers spoken or written communication, both off-line and on-line, such as chat, email and voice chat. These results are corroborated by the second study (see section 4.2).

Table 1. Generic group modeling activities

	Individual	**Social**	
		Coordination	**Cooperation**
Concretization	Understanding a situation	*Managing a modeling project*	*Clarifying an issue, Resolving a conflict*
Abstraction	Conceptualizing a situation	Communicating a view	Aligning views, Agreeing on a view

4.2 A student experiment

Version 1 of the tool was also tested on students in their last year studying informatics programs and taking a course on Information Systems and Business Processes. The course consisted of 17 students in 6 groups, most of which had 3 students. The aim of this study was primarily to investigate the usefulness and usability of the tool. All participants were required to fill out the respective questionnaire containing 13 pre-coded questions and 10 open questions. The former consisted of 4 questions each regarding usefulness, user interface and functionality, and one question concerning the mode that was used. The latter asked for concrete suggestions for improvements in these areas. Table 2 shows the average scores for the closed questions. A 5-point Likert scale was used ranging from *strongly disagree* (-2) to *strongly agree* (+2) for all but the last question.

Table 2. Usefulness and usability of the COMA tool

Category	**Score**
Tool improves modeling skills	1.06
Tool facilitates modeling	1.29
Tool supports personal way of working	0.76
Tool forces unproductive way of working	-0.82
Right number of windows	0.12
Too few windows	-0.63
Too many windows	1.29
All windows required	1.35
Sufficient edit functionality	0.94
Sufficient proposal functionality	1.09
Sufficient negotiation functionality	0.97
Sufficient functionality regarding latest version	1.09
Use of negotiation mode in %	63

The data in the table shows that the tool facilitates modeling and even contributes to an improved learning experience. The individual way of working is somewhat less strongly supported which suggests the need for further research on personal modeling styles. The user interface is mainly assessed with respect to the multi-window layout of the tool which was seen as the predominant feature. The participants claim that all windows are required for collaborative modeling but that they experience them at the same time as being too many. The answers to the open questions have helped us here to resolve this apparent contradiction (see below). The functionality of all components is seen as sufficient but the suggestions made in the open part show that there is still room for improvement.

The open answers reveal that the multiple-window dilemma can best be solved by presenting the windows in a different way, i.e. in the form of 3 alternative tabs instead of 3 simultaneously visible panes. Among the improvements mentioned most often were:

- Support for textual/audio communication in the group
- A larger working area in the editor component
- Being able to work off-line with the tool
- Keyboard shortcuts to mouse-operated functions

We have developed a new version of the COMA tool which addresses these concerns (see Fig. 3 for a screenshot).
The first bullet item addresses an issue that was also observed in the MobCom study, namely that some problems require a close social cooperation in the form of personal communication which was not supported by version 1 of the tool. Version 2 addresses this concern by incorporating a text and voice chat tool. The larger working area comes as a side-effect of the tab-control design of the new user interface. Off-line work is especially important in a student context where network access is not always available. Version 2 therefore allows modelers to work with locally available versions of files that have already been downloaded. The most commonly used functions are now also accessible via the keyboard.

Concerning the operating mode the quantitative results show that almost two thirds of the participants used the negotiation mode as we expected. They do so because it is a good way to reach an agreement and it puts all participants on an equal footing with nobody having undue influence. This mode was also experienced as a preferred way to approach a common view and for reaching consensus. But on the other hand more than one third used the facilitator mode which was originally intended for real-life modeling sessions where an external facilitator (e.g. consultant) is indeed present. The students, however, rather worked in small unfacilitated groups but found this mode useful anyway, mainly to speed up the modeling process because approval could be achieved without group consent.

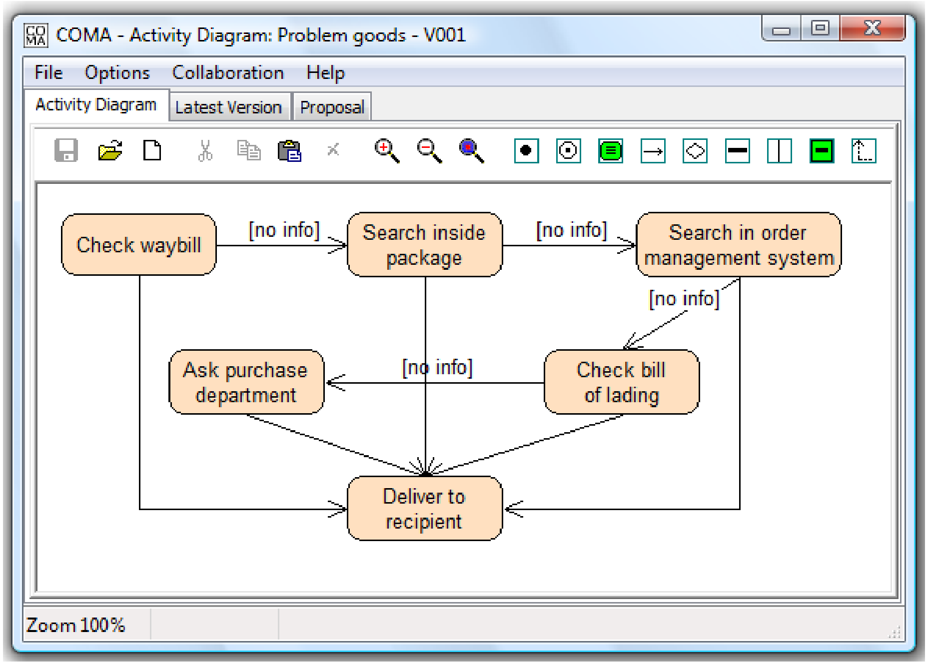

Fig. 3. Version 2 of the COMA tool

5 Conclusions

Based on earlier empirical work we have developed an architecture for collaborative modeling, COMA, and a tool that implements this architecture. Subsequent studies have showed that the tool fulfils the purposes for which it was built, i.e. modeling in groups is supported in a way that the group members experience as natural. The tool is also usable, i.e. the participants were able to handle it without effort in a short time. Most of the necessary functions were already available in the first version but some vital aspects, especially the layout of the user interface and the support of close social cooperation were observed as problematic/missing and were therefore changed/added in version 2.

In addition, the case study at MobCom has allowed us to identify a number of basic generic activities that are performed in group modeling to address important problems that arise in the social construction of a common artifact. We have analyzed them with respect to known factors from collaboration theory, namely the problem solving domain (natural vs. formal language) and the degree of collaboration (individual, coordination or cooperation). We thereby arrived at a table that allowed us to structure the identified generic activities in a systematic way. This has helped us in getting a better understanding of collaborative modeling, which in turn enabled us to assess the usefulness of COMA and to find gaps in the support that needed filling.

We believe that both the architecture and the tool are now in a sufficiently stable state to warrant a larger study. A possible focus of such a study is the integration of support for social activities on the natural language level.

References

1. Bommel, P.v., Hoppenbrouwers, S.J.B.A., Proper, H.A.E., Weide, T.P.v.d.: Exploring Modeling Strategies in a Meta-modeling Context. In: Meersman, R., Tari, Z., Herrero, P. (eds.): On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops - OTM Confederated International Workshops and Posters, AWESOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SebGIS, Proceedings, Part II, Montpellier, France, Vol. 4278. Springer, Berlin, Germany (2006) 1128-1137
2. Frederiks, P.J.M., Weide, T.P.v.d.: Information Modeling: the process and the required competencies of its participants. Data & Knowledge Engineering 58 (2006) 4-20

3. Hoppenbrouwers, S.J.B.A., Lindeman, L., Proper, H.A.: Capturing Modeling Processes - Towards the MoDial Modeling Laboratory. In: Meersman, R., Tari, Z., Herrero, P. (eds.): On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops - OTM Confederated International Workshops and Posters, AWESOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SebGIS, Proceedings, Part II, Montpellier, France, Vol. 4278. Springer, Berlin, Germany (2006) 1242-1252
4. Hoppenbrouwers, S.J.B.A., Proper, H.A., Weide, T.P.v.d.: Formal Modeling as a Grounded Conversation. In: Goldkuhl, G., Lind, M., Haraldson, S. (eds.): Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modeling (LAP`05), Kiruna, Sweden. Linköpings Universitet and Högskolan i Borås, Linköping and Borås (2005) 139-155
5. Persson, A.: Enterprise Modeling in Practice: Situational Factors and their Influence on Adopting a Participative Approach. Department of Computer and Systems Sciences, Stockholm University (2001)
6. Srinivasan, A., Te´eni, D.: Modeling as Constrained Problem Solving: An Empirical Study of the Data Modeling Process. Management Science 41 (1995) 419-434
7. Ericsson, K., Simon, H.: Protocol Analysis: Verbal Reports as Data. MIT Press, Boston (1993)
8. Stamper, R.: The Semiotic Framework for Information Systems Research. In: Nissen, H., Klein, H., Hirschheim, R. (eds.): Information Systems Research: Contemporary Approaches and Emergent Traditions. North-Holland, Amsterdam (1991) 515-517
9. Boehm, B., Grunbacher, P., Briggs., R.O.: Developing Groupware for Requirements Negotiation: Lessons Learned. IEEE Software 18 (2001) 46-55
10. Briggs, R.O., de Vreede, G.J., Nunamaker, J.: Collaboration Engineering with Thinklets to Pursue Sustained Success with Group Support Systems. Journal of MIS 19 (2003) 31-63
11. Conklin, J., Selvin, A., Buckingham Shum, S., Sierhuis, M.: Facilitated Hypertext for Collective Sensemaking: 15 Years on from gIBIS. In: Weigand, H., Goldkuhl, G., de Moor, A. (eds.): Proceedings of the 8th International Working Conference on the Language-Action Perspective on Communication Modeling (LAP'03), Tilburg, The Netherlands (2003)
12. Aiken, M., Vanjani, M., Krosp, J.: Group decision support systems. Review of Business 16 (1995) 38-42