Negotiating Models

Peter Rittgen

School of Business and Informatics, University College of Borås, 501 90 Borås, Sweden
peter.rittgen@hb.se

Abstract. We investigate the process of collaborative modeling by analyzing conversations and loud thinking during modeling sessions and the resulting models themselves. We discovered the basic activities of the modeling teams on the social, pragmatic, semantic and syntactic levels and derived a schema for the pragmatic level. Our main conclusion is that team-based modeling can be characterized as a negotiation process. Drawing on these results we suggest a tool support for modeling.

1 Introduction

With the abundance of literature on modeling one would expect that the process of modeling itself, i.e. how models are actually created, is well understood. And indeed, methods for developing, e.g., business process models can be found with ease [1-9]. Most of these methods operate on a coarse-grain level by specifying in which order diagrams should be developed, for example. Some provide also guidelines on how to create a specific diagram, especially in object-oriented modeling [10]. But in practice the use of a method is often reduced to the use of its notation as the former is fraught with principal problems [11]. We therefore claim that there is a need to study the modeling process from a descriptive rather than from a prescriptive perspective to find out what actually happens when people model. The ultimate aim is to support these activities with appropriate tools.

Descriptive approaches to understanding the modeling process are scarce. Only a few deal with collaborative modeling (see section 2). The others assume a scenario where a single expert modeler creates a formal model of some part of a business [12-15]. These studies identify sets of general heuristics for successful modeling without going down to the level of the concrete steps that are performed in creating models. Their results are hardly applicable to business modeling in general for a number of reasons. Firstly, a business model is rarely developed by an expert alone but rather by a team involving representatives of the respective business(es) and externals. Secondly, the problem domain of general business modeling is often less well-structured and formal languages are of limited use. Thirdly and last, the goal of providing tool support for collaborative modeling requires the identification of detailed steps.

The objective of this paper is to discover the elementary activities and the structure of the modeling process, i.e. a meta-model of the modeling process. This is done by studying, in a descriptive way, the work performed by small groups of modelers that were assigned the same task: To develop business process models for a hospital based

J. Krogstie, A.L. Opdahl, and G. Sindre (Eds.): CAiSE 2007, LNCS 4495, pp. 561–573, 2007.

on a detailed description of the processes in natural language. The group members were homogeneous concerning their modeling experience and their roles, i.e. there was no a-priori assignment of a group leader or modeling expert. The scope of this research is therefore limited to text-based modeling.

2 Related Research

Collaborative modeling processes have been studied by [16-20]. In [16] modeling involves domain experts, modeling mediators and model builders. It is viewed as a form of information gathering dialogue where knowledge is elicited from the domain experts. This view can be challenged because modeling is a social and communicative process where much of the information is created by and through the process rather than gathered from domain experts. We have therefore studied a situation where the participants had no a priori roles but all started from a similar position of having very little domain knowledge and collectively tried to make sense of the case described in a text document. [16] goes on developing meta model-based strategies which are of a prescriptive nature. Contrary to that and as outlined above our aim is to investigate what is actually done during modeling, i.e., we take a descriptive approach.

[17] emphasizes the importance of natural language as the primary medium and identifies two principal activities and associated roles: the domain expert who concretizes an informal model and a system analyst who abstracts a formal model. A detailed process model of both activities is given that again aims at prescribing steps to be performed to achieve a “good” analysis model. [18] distinguishes between an elicitation and a formalization dialogue and develops a modeling procedure by generalizing existing procedures for specific modeling languages. The authors claim that these procedures are descriptions of the modeling process (i.e., “documented procedures”) but the focus is again clearly on prescription (see, e.g., their use of the term “guidebook”). In this sense all other approaches can also be seen as descriptive but we use this term in a more direct sense, i.e. meaning the direct observation of modeling behavior with the purpose of getting a richer and more detailed description of the modeling process.

[19] 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 [16]). We fully agree that modeling is a conversation but we claim that it is a specific type of conversation, namely a negotiation. This idea is implicitly present in [19] where the dialogue structure contains negotiation elements such as propose and accept. We elaborate this point in the following sections and deliver a more detailed negotiation model. [19] 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.

[20] 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.

3 Research Method

Keeping this background in mind, we set out to study a situation where groups of modelers worked on a textual description of a business case with the purpose of deriving business process models. To understand the modeling process, we assumed that two factors are predominant in model creation:

- The internal mental processes of each modeler, and
- The conversations between modelers and within the group.

To get access to the former we used a think-aloud process-tracing methodology [13, 21] where the observants speak out what they are currently thinking. The utterances were then transcribed yielding the think-aloud protocols. The same is done with the conversations. In addition to that we also considered the product of the modeling process, the models themselves, to fill the gaps in the protocols and to help with interpreting ambiguous phrases in them. Open issues that could not be dealt with in this way were marked on the coding scheme and clarified by ex-post interviews with the respective groups.

To develop a preliminary coarse-grain categorization we turned to theories in the pertinent literature, particularly in organizational semiotics. We used the upper four 'rungs' of the semiotic ladder [22]: syntactic, semantic, pragmatic, and social. They refer to the structure of sign systems (e.g., a language), the meaning of the signs, their use, and the norms of a community, respectively. An initial coding phase within this framework revealed that the syntactic and semantic levels, which together make up the language level, are divided into the natural language domain and the modeling language domain depending on the kind of language used to describe the business.

The activities on the pragmatic level were classified as 'Understanding' and 'Organizing the Modeling Process'. The former term was then further refined into 'Undestanding the language' and 'Understanding the text', the latter can be divided into 'Setting the agenda' and 'Negotiation'. The social level consists of rules for acceptance and rejection in the negotiation. A detailed discussion of these categories can be found in the respective sections. The results are summarized in fig. 1.

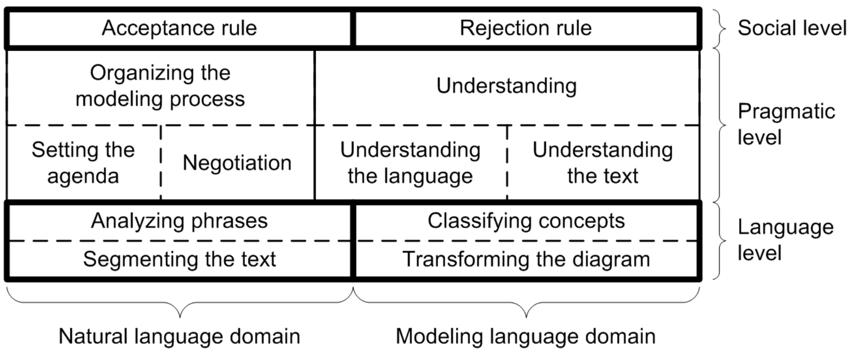

Fig. 1. Levels and domains

We conducted 3 experiments that involved a total of 26 groups of 2-3 students in informatics over a period of 3 years. The students were provided with a textual description of four business processes in a hospital. They were asked to model these processes with the help of two different modeling languages that they could choose freely from a set of four languages: ARIS-EPC [8], FMC-Petri nets [23], UML [24, 25], and DEMO [26]. Based on the results of these experiments we derived a layered meta-model of the modeling process that includes a model of the negotiation process.

4 Results

We carried out the main coding of the material within the framework stipulated by fig. 1. Examples of that procedure are shown in the respective section. The results are presented here in the order of the levels from top to bottom.

4.1 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.

4.2 Pragmatic Level

On the pragmatic level we discovered two distinct types of behavior, each of which can be classified in two sub-categories (the abbreviations of the categories are used as indices of the respective coded terms later on):

- *Understanding*, which concerns the text of the case description (index UT) or the (modeling) language (index UL), and
- Organizing the modeling process, which involves two types of activities: *setting the agenda* (index SA) and *negotiation* (index N).

Understanding was established by questions and answers. If the respondent could not provide clarification, an assumption was made. For details see table 1.

Agendas have been used by the participants in our study as an instrument for roughly structuring the modeling session. They were introduced in the beginning and then adapted during the session if necessary. On the whole most groups started by reading the case description completely and then organized their work around the flow of the text. For further details refer to table 1.

Table 1. Generic activities on the pragmatic level

Activity	Coding	Example
Modeler *m* makes proposal *p*.	$\textbf{propose}_N$ (*m*, *p*)	"I suggest that …"
Modeler *m* withdraws his/her proposal *p*.	$\textbf{withdraw}_N$ (*m*, *p*)	"My idea does not work; let us forget it."
Modeler *m* expresses consent to proposal *p*.	$\textbf{support}_N$ (*m*, *p*)	"I agree with that."
Modeler *m* expresses objection to proposal *p*.	$\textbf{challenge}_N$ (*m*, *p*)	"I think you are wrong."
Modeler *m* delivers argument *a* to support *p*.	$\textbf{argue\_for}_N$ (*m*, *p*, *a*)	"Yes, because that's an operation you do, so that must be a transaction."
Modeler *m* delivers argument *a* to challenge *p*.	$\textbf{argue\_against}_N$ (*m*, *p*, *a*)	"We cannot do this because an and-connector cannot have two inputs and two outputs at the same time."
Modeler *m* proposes *p′* instead of *p*.	$\textbf{counter}_N$ (*m*, *p*, *p′*)	"We should have the records after the evaluation and not at the same time."
Modeler *m* needs clarification on issue *q*.	$\textbf{ask}_{UT/UL}$ (*m*, *q*)	"Does the patient just come to the hospital?" (UT)
Modeler *m* provides a possible answer *a* to question *q*.	$\textbf{assume}_{UT/UL}$ (*m*, *q*, *a*)	"I think the sticky labels are required for tests."
Modeler *m* gives a definite answer *a* to question *q*.	$\textbf{clarify}_{UT/UL}$ (*m*, *q*, *a*)	"An XOR connector cannot follow an event." (UL)
Add activity *a* to the agenda as item number *n*.	$\textbf{add}_{SA}$ (*a*, *n*)	"First we will read the case completely."
Perform next activity on agenda.	$\textbf{perform}_{SA}$	"Next is patient care"

The majority of the activities on the pragmatic level were associated with negotiation, though (see also table 1). This is surprising as modeling is typically rather pictured as an intuitive act that is largely the product of a creative brain (e.g., a consultant) that possibly receives some input from other stakeholders in the modeling process (e.g., domain experts from the respective departments).

From these results we can draw interesting conclusions for the design of a system that supports modeling (see section “Architecture of a Modeling Support System”. This concludes the pragmatic level. The next section proceeds with the semantic level.

4.3 Semantic Level

The semantic level is concerned with the concepts of the modeled domain, in our case business processes. It is therefore also called the conceptual level. These concepts are

Table 2. Generic activities and concepts on the semantic level

Activity/concept	Coding	Example
Phrase t_1 is considered equivalent to phrase t_2.	**interpret**$_{AP}$ (t_1, t_2)	“An anamnesis is the same as a case history of the patient.”
Phrase t is considered an instance of foundational concept c.	**classify**$_{CC}$ (t, c)	“The ‘nurse’ is an actor.”
Actor x performs action a.	**perform**$_{CC}$ (x, a)	“The nurse treats the patient.”
Action a is triggered by the conjunction of events e_1, e_2, …	**wait-for**$_{CC}$ (a, e_1, e_2, …)	“When the certificate of discharge and the transfer report are ready, the nurse copies the transfer documents.”
Event e is raised by all of the actions a_1, … being finished.	**define**$_{CC}$ (e, a_1, …)	“A document is ready when it has been filled in and signed.”
Action a is triggered by any of the events e_1, e_2, …	**merge**$_{CC}$ (a, e_1, e_2, …)	“If the patient is new or the treatment is ineffectual, the physician examines him.”
Event e is raised by any of the actions a_1, a_2, … being finished.	**merge**$_{CC}$ (e, a_1, a_2, …)	“If the nurse has recorded the patient’s data or the clerk has registerered him, the patient is admitted.”
Actions a_1, … are triggered by event e simultaneously.	**trigger**$_{CC}$ (e, a_1, …)	“The arrival of the patient starts the admission process.”
Action a_2 follows action a_1.	**after**$_{CC}$ (a_1, a_2)	“First, the nurse hands the physician the admission papers, then he performs the admission examination.”
The completion of action a raises any of the events e_1, e_2, …	**branch**$_{CC}$ (a, e_1, e_2, …)	“The nurse determines whether the admission time is before 3 p.m. or not.
The completion of action a raises all of the events e_1, e_2, …	**fork**$_{CC}$ (a, e_1, e_2, …)	“When the examination has been completed, the tentative diagnosis is available and lab tests are requested.”
Action a creates info object o.	**create**$_{CC}$ (a, o)	“During registration a patient’s data is recorded.”
Action a removes info object o.	**remove**$_{CC}$ (a, o)	“Discharge implies deletion of the entry in the bed register.”
Action a uses info object o.	**use**$_{CC}$ (a, o)	“To assign a bed the nurse consults the bed register.”
Action a changes info object o.	**update**$_{CC}$ (a, o)	“The ward books are updated.”

often closely linked to the ones that are found in the language used for modeling the domain. The modeler expresses the perceived or constructed reality in terms of the concepts that the language provides, be it a natural language or a modeling language. This implies that the chosen language both enables and restricts the modeler in having and expressing certain thoughts. To prevent that these restrictions affect the number and types of concepts we identified in the modeling language domain (= *Classifying Concepts*, index *CC*) we treated them regardless of the language they were expressed in. As a consequence the results of the coding are not divided by language. Not all of the generic concepts are found in all languages, though. The foundational concepts are: actor, action/function, event and (information) object. Between them relational concepts are defined. They are listed in table 2 without the surrounding **classify** activity, for convenience. By phrase we mean a text fragment (e.g., from the case description). The index *AP* refers to *Analyzing Phrases*, the main activity of the natural language domain).

This completes the activities on the semantic level. The next section deals with the syntactic level.

4.4 Syntactic Level

On the syntactic level we distinguish again between the natural language and the modeling language domain. In the latter the diagrams of the respective language are built (*Transforming Diagrams*, *TD*). They consist of nodes, edges that connect them, and labels attached to both. In general the foundational concepts are represented by nodes and the relational concepts by edges but this is not necessarily true for all languages (e.g., Petri nets). In the natural language domain the text is segmented into useful units for the analysis on the semantic level (*Segmenting the Text*, *ST*). The generic activities on this level are listed in table 3.

Table 3. Generic activities on the syntactic level

Activity	Coding	Example
Text fragment t is used as the unit of analysis. The fragment can be: text, section, sentence, nominal phrase, verbal phrase or word.	$\textbf{focus}_{\text{ST}}$ (t)	“What does this sentence mean?”
Introduce a new node n of type t.	$\textbf{introduce}_{\text{TD}}$ (n, t)	“We insert an ellipse for the nurse.”
Attach label l to node n.	$\textbf{label}_{\text{TD}}$ (n, l)	“The function is called ‘Transport patient’.”
Remove node n.	$\textbf{remove}_{\text{TD}}$ (n)	“This place is not needed.”
Connect node n_1 to node n_2 with edge e of type t.	$\textbf{connect}_{\text{TD}}$ (n_1, n_2, e, t)	“We need a dashed line from ‘Nurse’ to ‘Update ward books’.”
Attach label l to edge e at place p.	$\textbf{label}_{\text{TD}}$ (e, l, p)	“Write ‘Before 3 p.m.’ on the arrow.”
Remove edge e.	$\textbf{remove}_{\text{TD}}$ (e)	“The arrow should not point that way.”

This completes the list of levels. In the following section we show an example of how a part of a model is negotiated.

5 Example

Somewhere in the middle of the modeling session concerning the admission process of the hospital, group B encountered a difficult situation. They were just focusing the following sentence in the case description (the part in square brackets is intended for the reader and was not present in the case text):

"The lab results are evaluated and the results [of the evaluation] are put in the medical record."

Table 4 shows the associated discussion of the group and the respective coding. The members of this group are called A and B. We have left out the syntactic level for convenience.

Table 4. An example discussion

Utterance	Coding
A: "I think we should introduce a function 'Evaluate lab results' first."	p_1 = **propose**$_N$ (A, **classify**$_{CC}$ ('Evaluate lab results', function))
B: "O.K."	**support**$_N$ (B, p_1), **accept** (p_1)
B: "And then another function 'Put results in medical record'."	p_2 = **propose**$_N$ (B, **classify**$_{CC}$ ('Put results in medical record', function))
A: "Wait! Is that not rather an output of the first function?"	p_3 = **counter**$_N$ (A, p_2, **classify**$_{CC}$ ('Medical record', object) + **update**$_{CC}$ (Evaluate lab results, Medical record))
B: "You are right! That makes more sense."	**support**$_N$ (B, p_3), **accept** (p_3), **reject** (p_2)

This small example is supposed to give an impression of how the modeling process is structured. In the next section we discuss a potential tool support for this process.

6 Tool Support for the Modeling Process

Our analyses of the modeling sessions showed us that modeling is a complex process involving issues such as collective sense-making, negotiations and group decisions. It is therefore worthwhile to consider tool support for this process. This is particularly true in an interorganizational setting where participants are often geographically distributed. The tool we envision helps group members in understanding the modeling situation, creating and discussing modeling alternatives, and deciding on the best one, all in a shared internet-based environment. The following paragraphs elaborate on the components that such a tool should provide.

According to our results modeling is a relatively well-structured process. It consists of a limited number of well-defined activities on all levels of the semiotic ladder. We are aware that further research will reveal more activities but from the experience of the three experiments that yielded a decreasing number of new ones, we are confident that the total number of activities will converge with respect to a given domain, e.g., business processes. The activities identified so far can therefore be assumed to be relatively stable in that domain. To a certain extent this is even true across different business process modeling languages, although the terminology of concepts may vary and not every concept is realized in each of the languages. But the findings will not carry over to another domain due to the domain specificity of the language level. The other levels are likely to work, though.

An analysis of the workflows on the pragmatic level revealed a structure that goes beyond the mere identification of generic activities. We found out that the negotation process actually follows a certain pattern. This pattern is shown in fig. 2.

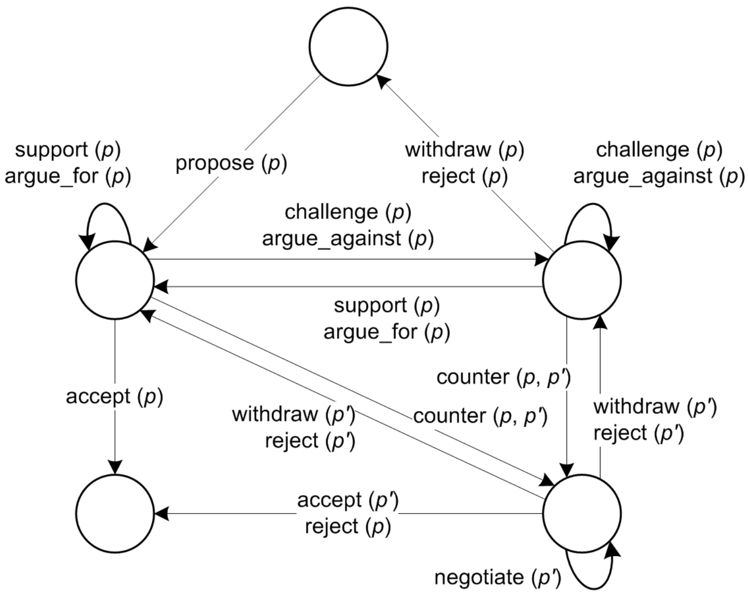

Fig. 2. Negotiation pattern

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 fig. 2 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.

The architecture of a modeling support system, 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 [18, 27-29] which is an important part of the modeling process. [29] 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 modeling support system (MSS). 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 an MSS 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 Fig. 2 can function as an initial workflow template controlling such a negotiation component. Once instantiated the actual workflow can then be adjusted to the concrete modeling situation.

7 Conclusions

We studied group modeling sessions in detail, both regarding conversations between the group members and the mental processes within each individual. By doing this we derive a sub-categorization of the upper four levels of the semiotic ladder, generic activities of business process modeling at all of these levels and a negotiation pattern at the pragmatic level. On the basis of these results we suggest a tentative architecture of a system that supports group modeling. Our aim with this research is two-fold. On the one hand we want to develop a better understanding of the modeling process that has been largely neglected by researchers so far. Such an improved understanding can lead to better modeling methods and thereby ultimately to higher quality of models.

On the other hand we are also interested in providing computer support to those modelers that work in a group environment. Modeling is a highly demanding task that is further complicated by the dynamics of group work. Effective support is therefore essential, especially if some of the group members are inexperienced as is often the case in business modeling sessions, where typically a majority of the participants does not have any modeling background. But it is precisely this latter type of participant that contributes most to the actual design of the model with his or her knowledge of the relevant business domain. Both the speed and quality of the models can therefore benefit tremendously if we can manage to involve these people directly as modelers instead of relying on the bottleneck of the modeling expert for all communication within the group. The suggested tool support can accommodate this by giving the expert seniority (i.e., the right to make the final decision) and turning the domain experts into effective consultants that make proposals (thereby reversing the traditional roles in IT consulting).

The modeling support system can also be seen as a special kind of group decision support system (GDSS, [30]) if we consider that the accept and reject decisions in the negotiation process are the key to model design. There is significant empirical support for the claim that GDSS are beneficial [30-39], particularly for larger groups and/or complex tasks. Many of these benefits carry over to modeling support systems, e.g., reduced meeting time, higher quality of the decisions, broader involvement of all participants, higher effectiveness of decisions, etc.

Our research studied text-based modeling only. This is not a realistic scenario for practical modeling situations. We are confident though that our results are relevant for real-world modeling to some extent. The social level is fairly independent of the way in which a modeling alternative was derived (text-based or other) as the decision rule rather depends on the alternatives themselves. The same is true for the language level as we can safely assume that natural language and modeling languages will play an important role in any modeling endeavor. We therefore expect differences primarily on the pragmatic level, and here especially in the areas “setting the agenda” and “understanding”. Whether modelers just interpret a text or communicate with domain experts will have considerable impact on the way the agenda is determined. Likewise the issue of understanding has to be extended to cover forms of communication other than analyzing text.

So far we have only looked at business process modeling. Other domains in the business and information systems areas remain to be explored. It should also be noted that our study has been performed in a contrived setting albeit with a realistic case. Further confirmation, and especially consolidation, is therefore required, preferably by means of a field study. In addition to this it seems reasonable to build a prototype of a modeling support system, and to test it in a realistic modeling scenario. We are confident that these measures will contribute to a better understanding of the process of modeling, both from a cognitive and a collaborative perspective, and they will eventually help us to better support modelers in their challenging task.

References

1. AMICE: CIMOSA: Open System Architecture for CIM, 2nd revised and extended version. Springer, Berlin (1993)
2. Barker, R.: CASE*Method: Entity-Relationship Modelling. Addison-Wesley, Wokingham (1990)
3. Bernus, P., Nemes, L.: A Framework to Define a Generic Enterprise Reference Architecture and Methodology. Computer Integrated Manufacturing Systems 9, 179–191 (1996)
4. Goldkuhl, G., Röstlinger, A.: Joint elicitation of problems: An important aspect of change analysis. In: Avison, D.E., Kendall, J.E., DeGross, J.I. (eds.) Human, organizational and social dimensions of Information systems development. North-Holland, Amsterdam, pp. 107–125 (1993)

5. Menzel, C., Mayer, R.J.: The IDEF Family of Languages. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) Handbook on Architectures for Information Systems, pp. 209–241. Springer, Berlin (1998)
6. Ould, M.: Business Process Management: A Rigorous Approach. Meghan-Kiffer Press, Tampa, FL (2005)
7. Roboam, M., Zanettin, M., Pun, L.: GRAI-IDEF0-Merise (GIM): integrated methodology to analyse and design manufacturing systems. Computer-Integrated Manufacturing Systems 2, 82–98 (1989)
8. Scheer, A.-W.: ARIS - Business Process Modeling. Springer, Berlin (1999)
9. Williams, T.J.: The Purdue enterprise reference architecture. Computers in Industry 24, 141–158 (1994)
10. Bennet, S., McRobb, S., Farmer, R.: Object-Oriented Systems Analysis and Design. McGraw-Hill, Maidenhead (1999)
11. Introna, L.D., Whitley, E.A.: Against method: exploring the limits of method. Logistics Information Management 10, 235–245 (1997)
12. Morris, W.T.: On the Art of Modeling. Management Science 13, B-707–B717 (1967)
13. Srinivasan, A., Te´eni, D.: Modeling as Constrained Problem Solving: An Empirical Study of the Data Modeling Process. Management Science 41, 419–434 (1995)
14. Willemain, T.R.: Insights on Modeling from a Dozen Experts. Operations Research 42, 213–222 (1994)
15. Willemain, T.R.: Model Formulation: What Experts Think about and When. Operations Research 43, 916–932 (1995)
16. Bommel, P., Hoppenbrouwers, S.J.B.A., Proper, H.A.E., Weide, T.P.v.d.: Exploring Modelling Strategies in a Meta-modelling 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. vol.4275, pp. 1128–1137. Springer, Berlin, Germany (2006)
17. Frederiks, P.J.M., Weide, T.P.v.d, Weide, T.P.v.d.: Information Modeling: the process and the required competencies of its participants. Data. & Knowledge Engineering 58, 4–20 (2006)
18. 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. vol.4275, pp. 1242–1252. Springer, Berlin, Germany (2006)
19. Hoppenbrouwers, S.J.B.A., Proper, H.A., Weide, T.P.: v.d.: Formal Modelling 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 Modelling (LAP'05), Kiruna, Sweden. Linköpings Universitet and Högskolan i Borås, Linköping and Borås, pp. 139-155 ((2005)
20. Persson, A.: Enterprise Modelling in Practice: Situational Factors and their Influence on Adopting a Participative Approach. Department of Computer and Systems Sciences, Stockholm University (2001)
21. Ericsson, K., Simon, H.: Protocol Analysis: Verbal Reports as Data. MIT Press, Boston (1993)

22. 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, pp. 515-517 (1991)
23. Keller, F., Wendt, S.: FMC: An Approach Towards Architecture-Centric System Development. In: Keller, F., Wendt, S. (eds.) 10th IEEE Symposium and Workshop on Engineering of Computer Based Systems, pp. 173–182. IEEE Computer Society, Pasadena, CA (2003)
24. OMG: UML 2.0 Superstructure Specification. OMG, Needham, MA (2004)
25. OMG: Unified Modeling Language: Infrastructure. OMG, Needham, MA (2006)
26. Dietz, J.L.G.: Understanding and modeling business processes with DEMO. In: Akoka, J., Bouzeghoub, M., Comyn-Wattiau, I., Métais, E. (eds.) Proceedings of the 18th International Conference on Conceptual Modeling ER ’99, pp. 188–202. Springer, Berlin (1999)
27. Boehm, B., Grunbacher, P., Briggs, R.O.: Developing Groupware for Requirements Negotiation: Lessons Learned. IEEE Software 18, 46–55 (2001)
28. 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, 31–63 (2003)
29. 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)
30. Aiken, M., Vanjani, M., Krosp, J.: Group decision support systems. Review of Business 16, 38–42 (1995)
31. Bamber, E.M., Watson, R.T., Hill, M.C.: The effects of group support system technology on audit group decision-making. Auditing: A Journal of Practice & Theory 15, 122–134 (1996)
32. Benbasat, I., Lim, L.H.: The effects of group, task, context, and technology variables on the usefulness of group support systems: A meta-analysis of experimental studies. Small Group Research 24, 430–462 (1993)
33. Bidgoli, H.: A new productivity tool for the 90’s: Group support systems. Journal of Systems Management 47, 56–62 (1996)
34. Burke, K., Chidambaram, L., Lock, J.: Evolution of relational factors over time: A study of distributed and non-distributed meetings.In: Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences vol. 4, pp. 14–23 ((1995)
35. Cass, K., Heintz, T.J., Kaiser, K.M.: Using a voice-synchronous GDSS in dispersed locations: A preliminary analysis of participant satisfaction. In: Proceedings of the Twenty-Fourth Hawaii International Conference on System Sciences vol. 3, 555-563 (1991)
36. Chudoba, K.M.: Appropriations and patterns in the use of group support systems. Database for Advances in Information Systems 30, 131–148 (1999)
37. Fjermestad, J., Hiltz, S.R.: An assessment of group support systems experiment research: Methodology and results. Journal of Management Information Systems 15, 7–149 (1998/1999)
38. Jackson, N.F., Aiken, M.W.V., Mahesh, B.H., Bassam, S.: Support group decisions via computer systems. Quality Progress 28, 75–78 (1995)
39. Townsend, A.M., Whitman, M.E., Hendrickson, A.R.: Computer support system adds power to group processes. HRMagazine 40, 87–91 (1995)