COLLABORATIVE BUSINESS AND INFORMATION SYSTEMS DESIGN

Peter Rittgen, Vlerick Leuven Gent Management School, Belgium, and University College of Borås, Sweden

ABSTRACT

Collaborative business and information systems design touches a number of issues that lie within the realm of different research areas. It deals with design as such and in particular with design in and for groups. It is also concerned with socio-technical systems and hence with human-computer interaction as well as IT-mediated human-human interaction. This introduces collaboration issues. The significant complexity of the business and information systems that are in the focus of the design endeavor calls for modeling as an instrument for managing this complexity. This paper maps the terrain of collaborative business and information systems design by surveying the contributions that are made by related areas of research.


INTRODUCTION

Designing anything – whether a simple object of daily use or a complex information system – is a challenging task. It requires creativity, courage, inventiveness and a sense for innovation. In the case of businesses and their information systems the situation is further complicated. On the one hand they determine each other, which makes it impossible to design or study them in isolation. On the other hand these systems are collaborative systems, i.e. human beings work together with others and/or computerized systems to fulfill business objectives. This suggests that the design of such systems also has to be a massive collaborative effort that involves contributions from a large amount of stakeholders with different backgrounds: project managers, domain experts, information technology experts, consultants, executives, and so on.

Apart from design and collaboration there is a third aspect that plays an important role. The complexity of business and information systems is such that building them requires a succession of abstraction layers, each of them more concrete than the preceding one, until a level is reached that can actually be realized. Each of the layers is typically represented as some kind of model. Modeling is therefore also an issue that needs to be considered.

The following figure (Fig. 1) illustrates the three aspects of collaborative business and information systems design (CBISD). It shows that all three overlap each other with CBISD in the middle. So far a substantial body of research exists concerning the pair wise intersections. The following sections elaborate on that.
The following sections are structured as follows. We first introduce the dimensions of collaborative design in section 2. These dimensions are a useful instrument for the classification of collaborative design problems. But they can also help us in finding out which type of solution fits to which type of problem, i.e., in understanding the characteristics a solution must exhibit in order to solve the respective problem.

Collaboration issues have been studied thoroughly in a field that is called computer-supported cooperative work. Many of the methods and techniques from this field have been used in the collaborative design of business and information systems so we will take a closer look at them in section 3.

As already mentioned, the design of business and information systems requires levels of abstraction to manage the inherent complexity. Modeling as a discipline provides the tools, e.g., modeling languages and methods, to handle each abstraction level. We therefore focus on aspects of collaborative modeling in section 4.

Section 5 briefly outlines approaches to collaborative and model-driven design. The former deals with issues such as participatory design and user-centered design, the latter with the model-drive architecture of software design.

**DIMENSIONS OF COLLABORATIVE DESIGN**

We distinguish three dimensions of collaborative design: Type of group, abstraction level and degree of participation (see Fig. 2). The parameter type of group relates to the homogeneity of the design group. In a homogeneous group all participants have a similar background and share common knowledge with respect to the design problem. An example of such a group is a number of architects that work on the design of a house. They have the same education and they master the common design language of architectural drawings. They also share knowledge with respect to the restrictions of construction as, e.g., imposed by the laws of statics. Their collaboration therefore takes place in a well-defined arena which facilitates the development of tools and methods that support the design process.
In collaborative business and information systems design the groups are typically of a heterogeneous nature. This makes it more difficult to find suitable design languages and it also affects consensus building. Nevertheless the methods and techniques that were made for homogeneous group can provide a basis to develop support for more disparate teams of people. For example, collaborative graph editors can be embedded in a collaborative modeling environment for heterogeneous groups.

The second dimension, abstraction level, relates to the number of design elements that are not considered. A low abstraction level means that we are close to the concrete artifact we design and few or no details are left out. An artist who is making a realistic sculpture of a horse works on a low abstraction level, for example. She considers all the details of a horse to make her sculpture as realistic as possible. The next higher level of abstraction is reached when the designer makes a model or a sketch before working on the actual object.

In some areas the modeling language is highly formalized and allows for a straightforward transfer of the model into the artifact. In these areas we tend to identify the model with the design. We say that the engineer designed the car and mean that he drew the model which served as a blueprint for the building of the car. We do not see the engineering activity as a modeling activity. Computer-support for collaborative design in these areas is therefore less of a challenge (see respective section).

When dealing with information systems we typically employ a number of abstraction levels. On the highest level we leave out most of the details of the system and focus only on the users of the system and its basic functionality. Such a model, e.g. a Use Case Model of the Unified Modeling Language (UML), cannot be translated into an application automatically. Instead a designer has to develop a Class Diagram, which then will be complemented by other more detailed UML diagrams, which finally will be the basis for programming.

Collaborative business and information systems design starts at a high abstraction level and proceeds via more detailed levels to the lowest abstraction level, i.e. the concrete socio-
technical system. Again, the lessons learned from low-level methods and languages are useful for the development of many-level approaches.

The third dimension, degree of participation, refers to the role that the stakeholders have in the design process. A low degree means that their role is rather passive, a high degree relates to a more active role. An example of the former is facilitated business process modeling where the facilitator consults the participants by asking questions that serve the purpose of eliciting the information that is required for building the model. Participants cannot directly shape the model and are often not able to assess whether the model correctly represents their views.

An example for a high degree of participation is brainstorming where group members can freely suggest any idea that comes to their mind and they are also involved in the organization and ranking of these ideas.

It can be safely assumed that the collaborative design of business and information systems requires a high degree of participation in order to ensure stakeholder buy-in. The section on collaborative modeling discusses this issue in greater detail.

**COMPUTER-SUPPORTED COOPERATIVE WORK**

Under the heading “computer-supported cooperative work” a number of systems have been studied that help groups in coordinating their work. These systems range from simple email applications to sophisticated groupware.

The more advanced CSCW systems were usually tailored for the support of specific tasks. A common task frequently studied was that of group meetings with the purpose of generating and organizing ideas (i.e., brainstorming sessions). Groupware systems that support such sessions are called electronic meeting systems (EMS).

Another major task focused in CSCW research is that of decision support. Because of the importance of decisions in business the research field of group decision support systems (GDSS) covers a broad range of publications.

Negotiation support systems (NSS) are less frequently discussed although they also play a vital role in business. Section 3.3 surveys these systems with respect to their suitability for collaborative business and information systems design.

**Electronic Meeting Systems**

EMS’s are computer systems that are designed to support electronic meetings. They are also called group support systems. Their foundations lie in the Team Theory of Group Productivity (Briggs, 1994) that identifies three cognitive processes in group members: communication, deliberation, and information access that all compete for the group member’s attention thereby reducing group productivity. EMS’s have been found to decrease the demand for attention and increase productivity. These processes constitute the horizontal dimension of the so-called Groupware Grid (Nunamaker, Dennis, Valacich, Vogel, & George, 1991).

The vertical dimension is the degree of participation already mentioned above but in the EMS literature it is called levels of group effort. On the individual level there is no participation in group work. All group members work independently, albeit towards a common goal. On the coordinated level members still work independently but their efforts are coordinated in some way, e.g. by exchanging documents. On the group dynamics level there is a concerted effort, i.e.
the team members work closely with each other (in our terminology this is called collaboration).

EMS’s provide support in all of the nine cells of the grid but the focus is often on the group dynamics level. Typical functions of an EMS are:

- Electronic brainstorming: Generating ideas
- Categorizing: Mapping ideas to pre-defined categories
- Outlining: Organizing ideas in a tree-like hierarchy
- Rank order voting: Sorting ideas by preference
- Multi-item, multi-criteria polling: Evaluation of ideas and consensus building

Most EMS’s support collocated meetings in a so-called decision room but there is also a line of research that deals with distributed group work (Kim, Hiltz, & Turoff, 2002). The proponents of electronic meeting support claim that EMS’s have a positive effect on the outcome of meetings. There is general consent to the overall claim but there is some debate as to the conditions for successful EMS use. Most researchers agree that electronic meeting systems do not have a direct impact on meeting results. They rather moderate the meeting process, which in turn influences the output of the meeting (Reinig & Shin, 2002).

There are two mechanisms by which EMS’s can make a difference over face-to-face meetings: anonymity and simultaneity. Anonymity means that utterances cannot be attributed to their originator. It is assumed that this will lead to additional ideas and comments that the proponents would otherwise not have felt comfortable to share. By simultaneity we mean the possibility of all team members to utter their comments and ideas at the same time. In conventional meetings only one person can speak at a time which leads to air time fragmentation and thereby to a loss of ideas.

Both anonymity and simultaneity are supposed to moderate group process gains (such as more objective idea evaluation, learning from others, etc.) and process losses (such as air time fragmentation, production blocking, conformance pressure, free riding, evaluation apprehension, etc.). These process characteristics are assumed to determine outcome factors such as efficiency, effectiveness, satisfaction, consensus and so on (Fjermestad & Hiltz, 2001).

Experimental results were inconclusive (Fjermestad & Hiltz, 1999) but case and field studies showed the contributions of EMS’s in reducing meeting time and increasing meeting productivity (Nunamaker, Briggs, Mittleman, & Vogel, 1997). The major lessons that can be learned from EMS research are:

- An EMS works best with larger groups.
- An EMS works better for larger problems that require repeated sessions.
- An EMS works best in synchronous, collocated environments (i.e. same time, same room). Distributed scenarios require dedicated coordination structures (Kim, Hiltz, & Turoff, 2002).
- An EMS cannot replace the facilitator.
- Prior training on the EMS is a prerequisite for successful use.
- An EMS supports divergence better than convergence (for the latter see group decision support systems).
The application of EMS to collaborative design, especially to group modeling has also been studied (Dean, Orwig, Lee, & Vogel, 1994; Dean, Lee, Orwig, & Vogel, 1994; Dean, Orwig, & Vogel, 2000). The mentioned studies found the same drastic time reductions and productivity enhancements as the other EMS studies but modeling was limited to the textual parts of the IDEF (Integrated Computer-Aided Manufacturing Definition) language.

In collaborative business and information systems design most of the conditions for the successful use of EMS prevail. In addition, design languages often use diagrams instead of text. While this may seem to make the use of EMS even more difficult and less effective, it is actually a point in favor of their use. The cognitive load associated with deliberating on an idea in the form of a diagram is higher than with an idea that is expressed as a textual phrase. The effort of communicating the idea behind the diagram verbally to the facilitator/modeler is likewise higher. An EMS can reduce this time by making the proposed diagram directly available to the facilitator/modeler. In this way the attention that the proponent has to spend on communicating is freed and can be used on the more productive deliberation instead. The section on collaborative modeling elaborates on that.

Group Decision and Negotiation Support Systems

The literature on group decision support systems (GDSS) largely coincides with that on EMS. The reason for this lies in the fact that both are studied under the heading “group support systems” (GSS). This label is misleading because it suggests that the same support system can be used for all group work regardless of the task that the group is assigned. We believe that the tasks of idea generation and decision-making, for example, are sufficiently unlike to warrant a different kind of support.

Not surprisingly, empirical studies have found that EMS’s perform well on ideation and similar divergent tasks, but less so on convergent tasks such as decision-making (Fjermestad & Hiltz, 1999). Although EMS’s do possess the required tools (e.g. for ranking, voting and polling), their principle architecture does not lend itself to the decision-making process.

For example, anonymity, a valued characteristic in idea generation, can be obstructive in decision-making where a position needs to be defended personally to lend it weight and to convince others. Simultaneity, likewise, no longer saves precious air time but destroys the logical flow of an argument. It is true that anonymity and simultaneity can be “switched off”. But as they are defining characteristics of an EMS, taking them away leaves us with nothing.

In a broad sense decision-making consists of the following activities:

1. Generating decision alternatives,
2. Determining the respective merits and drawbacks of each alternative,
3. Assessing each alternative in the light of the pros and cons, and
4. Selecting one of the alternatives.

The first activity is clearly an ideation task which might well be supported by an EMS. The second is also divergent in the sense that it generates comments in the form of arguments for and against alternatives. With the third activity, now, we enter the convergent phase, the one which is typically associated with decision-making in the narrow sense. Here we judge the weight that each argument carries and the impact it has on the value of the respective alternative.

It is this judgment that ultimately allows us to narrow down on a final alternative. If the decision is that of an individual, only one preference system needs to be taken into account. For
example, a decision-maker might ask herself whether a pro of alternative $A$ weighs heavier than a pro associated with alternative $B$.

In group decisions we have at least two further issues that significantly complicate the decision-making process. Firstly, group members have different preference systems that might be partially conflicting. For example, one participant might think that an argument $A$ is a strong pro, while another one considers it a weak pro. Yet another group member might even see it as a con.

Secondly, the exchange of arguments might change the participants’ state of mind. Such a change of mind might affect an individual’s assessment of an alternative, or even his preferences. Theoretically this might lead to an endless circle of new arguments and new assessments but in practice the process usually converges. But even so it can draw out group decisions considerably.

The existence of different preference systems makes group decisions essentially a negotiation process. This has been acknowledged by research (Tavana & Kennedy, 2006) and consequently a number of approaches exist that take a closer look at negotiation support systems (de Moor & Weigand, 2004; Schoop, Jertila, & List, 2003; Schoop & Quix, 2001).

Complex group decisions are also required in collaborative business and information systems design where a choice between design alternatives must be made. As design alternatives are usually complex, e.g. they might cover a set of information models or even complete enterprise architectures, balancing their pros and cons in a group can be a tremendous challenge.

Existing negotiation support systems usually focus on formalized textual documents such as contracts. Design decisions go beyond this in involving both unstructured and graphical components. An example of the former is the negotiation of the meaning of the basic elements of the design language to facilitate mutual understanding within the group. This issue is also known as ontology negotiation and is treated in a number of publications (Bailin & Truszkowski, 2002; Diggleen, Beun, Dignum, Eijk, & Meyer, 2007). Tool support for this task has also been studied, e.g. recently in (Xexéo et al., 2005).

To address the second issue, graphical components, a negotiation support system for design decisions needs to provide mechanisms for comparing graphs parts, composing graphs, effecting traceable changes and additions, and so on. For each addition or change the system has to store information about who made it, why it was made, what are the pros and cons voiced by the group members, and whether it is endorsed by the group. It should also provide functionality to roll back changes that are not accepted. For each version of a design its complete history (earlier versions) and the arguments that led to the current one need to be accessible. Some of these issues are addressed in the section “Collaborative Modeling Architecture.”

**COLLABORATIVE MODELING**

Modeling is an important part of design. Most design-oriented disciplines make use of some kind of model before building the actual artifact. Architects make blueprints; engineers develop technical drawings; artists do sketches; and so on. Models play an even greater role when the artifact increases in complexity. In software development, for example, there are several layers of models required: requirements, scenarios, architecture models, information models, process models, communication models, interaction models, prototypes, and finally the production code, which in turn is an input model for the next release round.
Collaborative business and information systems design therefore also requires layers of models that codify the structure and interaction of human and computer agents at different levels of abstraction. Consequently there is substantial research dealing with the modeling of business and/or information systems. In most of this research there is an implicit assumption that modeling for such systems is performed in groups. But the majority of them do not address collaborative modeling explicitly by looking at group dynamics and interactions in the modeling process. Instead they focus on the method to be used, which often specifies modeling activities only at the group level but not at the individual level and particularly not at the relation between these two levels. Some methods are concerned with facilitation techniques that suggest certain procedures for one of the group members, i.e. the facilitator, but not for the others.

In this section we have compiled three approaches that take an explicit collaborative stance on modeling issues: Group Modeling, Participative Enterprise Modeling and Collaborative Modeling Architecture.

**Group Modeling**

In the nineties a group of researchers at the University of Arizona was convinced that the use of computer support would substantially improve group modeling (Dean, Lee, Orwig, & Vogel, 1994). They built three generations of group modeling support systems that were built on the existing of electronic meeting systems described above. They used the IDEF0 activity modeling language that describes a graph of activity nodes that are connected by ICOMs, i.e. definitions of Inputs, Controls, Outputs and Mechanisms. In spite of the graphical nature of the language the EMS-IDEF0 tool was essentially a collaborative text editor for model input. In the 3rd generation it was complemented by a graphical viewer that visualized the textual input “online” on a separate (!) workstation. Manipulation of the graph itself was not possible, i.e. graph editing was not provided for. The approach was later extended to a graphical business process language in the Collaborative Distributed Scenario and Process Analyzer (Lee, Albrecht, & Nunamaker, 2001).

Its obvious technical limitations notwithstanding, the tool proved very successful. This success was partly due to the fact that the tool allowed for simultaneous editing of different model parts, which provided for parallelization of the modeling effort that was especially beneficial for large groups and complex models. It should be noted though that only a few group members would actually work on the model at any time while the others would stand around the workstations and provide oral input. The tool is therefore not a fully collaborative tool where each participant would be equipped with computer support. We come back to this in the section “Collaborative Modeling Architecture.”

An important part of the success can also be attributed to the fact that the tool gave instant feedback to the “bystanders” who could immediately react to the visual display of the growing model by making comments and proposing changes. The impact of tool support on group modeling is dramatic according to (Dean, Lee, Orwig, & Vogel, 1994). The tool allows for handling groups of twice the size that traditional techniques can handle (20 vs. 10). Individual efficiency increased by 85 % in terms of activities, and group productivity went up by 251 %. The time spent on modeling is decreased by almost 70 %, making an EMS-supported project approximately 3 times faster than a traditional one (Dean, Orwig, Lee, & Vogel, 1994).

Quality has also increased in a number of measured dimensions but facilitation has an important influence here. (Dean, Orwig, & Vogel, 2000) developed a technique they call top-down, integrated (TDI) approach, which goes some way in showing how the internal work of a group should be organized to leverage the benefits of tool support and to ensure high-quality models.
The lessons that Dean et al. have learned from their studies are also important lessons for collaborative business and information systems design. The fact that increasing model complexity warrants tool support is certainly applicable here as business and information systems are complex socio-technical systems. But the EMS-IDEF0 tool does not provide any support for the negotiation part of collaborative modeling. For full support of the group modeling process we therefore need to combine a group modeling system of EMS-type with a negotiation support system (see the respective section) in an adequate way. We elaborate on this issue in the section “Collaborative Modeling Architecture” takes up this issue.

**Participative Enterprise Modeling**

Participative Enterprise Modeling (PEM) focuses on the group process itself. It is based on a number of case studies, three of which are described in (Stirna, Persson, & Sandkuh, 2007). An important aspect in PEM is the facilitation roles: Process owner, Facilitator, Modeling expert, Tool operator and Domain expert. The authors have used two different scenarios for modeling sessions. In one the tool operator was supported by a computerized tool. The other roles were not supported by a computer system but they were able to see the current status of the model that was projected onto a screen. All interaction regarding changes to the model had to be channeled through the tool operator who consequently represented a bottleneck in the modeling process.

The second scenario involved a plastic wall that was used to attach post-it notes to it representing the nodes of the diagram. This allowed all participants to take an active role as everybody could go to the plastic wall and fix a note there to make a proposal regarding the model. It is interesting to note that this scenario resulted in models of higher quality than the ones produced with tool support. On the face of it this seems to indicate that a simple plastic-wall support outperforms computer support.

But on closer inspection we can see a fundamental difference between the scenarios: The tool only supports the tool operator; the plastic wall supports all group members. A participant who wants to suggest something does not have to communicate her idea to somebody who then effects the change but can make it herself. The plastic wall is therefore rather a proof of the fact that active involvement of all participants is really the crucial point and that a computerized tool support should mimic the plastic wall in giving all group members direct computer access to make instantaneous proposals. But it should also provide features that go beyond the plastic wall to leverage other benefits of technology. The next section explores this issue.

**Collaborative Modeling Architecture**

The preceding sections have shown that is not enough to combine electronic meeting systems with modeling tools to support group modeling in an effective way. We also need negotiation support for the convergent part of modeling. But the lessons learned from participative modeling have shown us that this is still not enough. We must also make sure that all participants get actively involved to ensure that they embrace the result of the modeling process. A plastic wall makes it easy for group members to contribute and a collaborative modeling support system should allow for the same instead of just supporting the facilitator. The Collaborative Modeling Architecture (Rittgen, 2009) accomplishes that by studying interactions between group members during modeling from a social and pragmatic perspective. The result is an architecture for collaborative modeling comprising social norms, a negotiation pattern and core modeling activities.
The social norms within a modeling team are mainly made up of rules for determining whether a proposal is accepted or rejected. It has been 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 is applied occasionally to force a decision if a negotiation gets stuck, i.e., when there is no more changes in the individuals’ convictions over an extended period of time. There are two types of rules:

- Rules of majority, where a certain number of group members has 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 is sometimes specified (e.g., for the case of an equal number of supporters and opponents). A tie-break might involve issues of seniority.

- Rules of seniority, where the weight of a group member’s opinion is related to her status within the group. The status can be acquired (e.g., by experience) or associated with a position to which the member is appointed. A frequent example of this is the case of a more experienced modeler who is considered as the leader by the group and takes decisions on their behalf. Other members often fill the role of ‘consultants’ in such a case.

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

Analyses of core modeling activities on a pragmatic level revealed recurring activities that the group members engage in to solve their modeling problems. In general terms they can be summarized as follows:

- Create/change an individual model
- Discuss a proposal
- Propose a new/changed individual model to the group
- Comment on a proposal
- Vote on/assess a proposal
- Discuss an unclear issue
- Decide on a group model (new version)
- Merge proposals
- Discuss use of modeling language
- Reuse parts of a proposal/version

A substantial part of activities on the pragmatic level are associated with negotiation. This is surprising as modeling is typically rather pictured as an elicitation process that is combined with individual model construction by the facilitator/modeler. The results rather suggest that modeling is a process that combines idea generation with model negotiation.

The negotiation activities on the pragmatic level reveal a structure that goes beyond a set of generic activities. The negotiation process actually follows a certain pattern. This pattern is shown in Fig. 3.
Fig. 3. 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 this pattern the negotiation component of a modeling support system can be controlled. On other semiotic levels the pattern of activities strongly depends on the modeling language that is used, which restricts the potential for generalization. This will affect the kind of support a tool can provide at the language level.

The core modeling activities, the negotiation pattern and the social rules together are called the COllaborative Modeling Architecture (COMA). This means that COMA covers the social and pragmatics levels of the semiotic ladder. For the semantic and syntactic levels it relies on existing modeling languages. The details regarding the foundations of COMA can be found in (Rittgen, 2007b).

Functions supporting the core modeling and negotiation activities as well as the social rules are implemented in a tool for collaborative modeling. It is called the COMA tool and can be downloaded at http://www.coma.nu (Rittgen, 2007a).

Activities involving discussions are not supported. For them COMA relies on face-to-face conversations, i.e. it is assumed that the group members are located in the same room. If this is not the case, the group can avail themselves of standard Voice-over-IP teleconferencing for this purpose. Fig. 4 shows a screenshot of the COMA tool that was taken during a modeling session.
Fig. 4. COMA screenshot

It shows a snapshot of the modeling process at a certain stage. This is supposed to give the reader an example of how modeling in COMA proceeds and how it helps with a particular problem, namely that of “Making different views converge”. The group in question was concerned with the handling of so-called problem goods, i.e. goods with an unclear recipient. In a first step they simply wrote down all activities that are involved thus arriving at the first version (Eliciting individual views, upper pane). One member suggested to order the activities in a certain sequence and made a respective proposal (lower right pane). He knew from experience that this was indeed the order in which the activities were carried out at Goods receipt. Another modeler agrees with the principle sequence but he is quite sure that the search for the recipient is terminated as soon as the recipient is identified and further steps are skipped. He draws the respective diagram in his editor window (lower left pane) and makes a counter-proposal. On seeing the apparent conflict the first modeler confirms with the operations staff that this is indeed the case and withdraws his original proposal in favor of the new one. The new proposal received supporting votes by the other team members and was subsequently adopted by the group as version two.

The effectiveness of COMA in supporting group modeling has been demonstrated in a comparative empirical study (Rittgen, 2009). It found better model acceptance by the group, faster progress in modeling, less facilitator overload, improved convergence of views, increased perceived model quality, and better model comprehension in the tool-supported sessions.

COLLABORATIVE AND MODEL-DRIVEN DESIGN
An early approach to collaborative design is called Participatory Design. It originated in Scandinavia (Floyd, Mehl, Reisin, Schmidt, & Wolf, 1989) as an a way to achieve industrial democracy (Ehn & Kyng, 1987). Workers should be empowered by having more control over the conditions of their workplace. Participatory Design was seen as a means to this end in the area of information systems. But beyond political agendas user involvement has also tangible
benefits for the business. Users who participate in the design of an information system put up less resistance to its introduction and their work is likely to be supported more effectively.

But so far approaches to Participatory Design failed to deliver a comprehensive and rigorous methodology for systems design. Suggested methods were often restricted to a subset such as user interface design (Muller, 1992). User-Centered Design aims at overcoming these deficiencies by providing a set of methodologies for involving users in a number of system development activities. They include field studies (including contextual inquiry), user requirements analysis, iterative design, usability evolution, task analysis and so on (Mao, Vredenburg, Smith, & Carey, 2005). These methods are tailored to a potentially large and unstructured user base as typically found in areas as electronic commerce.

The term User-Centered Design is somewhat misleading though as the focus is not on the actual design phase of systems development but rather on requirements elicitation and testing. The former is still performed without user involvement mostly. The only exceptions to this are cases where the users are at the same time developers, e.g. as in open-source projects.

Successful collaborative design environments have also been established in fields that have a homogeneous user base. An example of that is Collaborative Computer-Aided Design (Saad & Maher, 1996). An important issue in engineering-oriented disciplines of design is collaborative graph editing (Pereira Meire, Borges, & Araújo, 2007).

While the term design is interpreted in a wide sense in collaborative design and often excludes design as a phase in systems development, the latter notion is addressed by the field of model-driven design. It is embodied in the Model-Driven Architecture (MDA) of software engineering (OMG, 2003). MDA acknowledges that the design of complex systems such as business and information systems requires layers of models with an increasing level of detail and it suggests that there should be precisely three such layers: Computation-Independent Model (CIM), Platform-Independent Model (PIM) and Platform-Specific Model (PSM).

Whether these three layers are sufficient in any case remains to be investigated. MDA proposes that the Unified Modeling Language (UML) (OMG, 2004) should be used at all levels. At least at the CIM level this is doubtful as the UML does not provide for a business modeling notation. There is so far also no comprehensive methodology that covers all levels. Inter-level transformations do exist though for specific pairs. Among the solutions for CIM-PIM pairs are (Ouyang, Dumas, Hofstede, & Aalst, 2007) for the Business Process Modeling Notation (BPMN) and the Business Process Execution Language (BPEL), and (Rittgen, 2006) for DEMO (Dynamic Essential Modeling of Organization) (Dietz, 2006) and the UML.

**CONCLUSION**

From what was said above we can conclude that the collaborative design of business and information systems is a vast field that is still largely unexplored. The important cornerstones of this field are its collaborative nature, active stakeholder involvement to ensure their buy-in and a suitable abstraction level to discuss and negotiate design with these stakeholders. Aspects of this problem have been addressed by related fields such as electronic meeting systems, group decision and negotiation support systems, group modeling, participative enterprise modeling, participatory design, user-centered design and model-driven architecture.

But none of these approaches covers collaborative business and information systems design to its full extent. The closest approach is that of collaborative modeling which combines modeling as idea generation, modeling as negotiation, group interaction and active participant involvement. But even here there is still a multitude of issues that remain to be explored such as
the negotiation of shared meaning, improving the motivation of participants, exploring different modes of collaboration and so on. This special issue presents recent approaches that advance our knowledge in this field by filling some of the gaps.

REFERENCES


