

Journal of the Association for Information Systems

J AIS

Special Issue

Collaboration Engineering: Foundations and Opportunities: *Editorial to the Special Issue on the Journal of the Association of Information Systems*

Gert-Jan de Vreede
Center for Collaboration Science
University of Nebraska at Omaha
and
Delft University of Technology
gdevreede@mail.unomaha.edu

Anne P. Massey
Indiana University
amassey@indiana.edu

Robert O. Briggs
Center for Collaboration Science
University of Nebraska at Omaha
rbriggs@mail.unomaha.edu

Abstract

Collaboration is a critical phenomenon in organizational life. Collaboration is necessary yet many organizations struggle to make it work. The field of IS has devoted much effort to understanding how technologies can improve the productivity of collaborative work. Over the past decade, the field of Collaboration Engineering has emerged as a focal point for research on designing and deploying collaboration processes that are recurring in nature and that are executed by practitioners in organizations rather than collaboration professionals. In Collaboration Engineering, researchers do not study a collaboration technology in isolation. Rather, they study collaborative work practices that can be supported on different technological platforms. In this editorial, we discuss the field of Collaboration Engineering in terms of its foundations, its approach to designing and deploying collaboration processes, and its modeling techniques. We conclude with a Collaboration Engineering research agenda for the coming decade.

Volume 10, Special Issue, pp. 121-137, March 2009

1. Introduction

In the knowledge economy, organizations frequently face problems of such complexity that no single individual has sufficient expertise, influence, or resources to solve the problem alone. Collaboration has, therefore, become a ubiquitous feature of organizational life. We define *collaboration* as joint effort toward a group goal. In many organizations, collaborative work practices such as strategic planning, software requirements negotiations, and marketing focus groups are now critical to survival and success. Collaboration, however, is a mixed blessing. With the value it creates, collaboration also brings its own special economic, socio-emotional, political, and cognitive challenges.

Since the 1980s, many researchers have explored ways in which collaboration technologies such as e-mail, web conferencing, and Group Support Systems can help organizations improve the productivity of their collaborative efforts. Despite many successes in the field, however, such technologies have not seen wide-spread implementation as they require that users have extensive knowledge about how to use technology to invoke, sustain, and change useful patterns of collaboration.

Collaboration professionals, such as expert facilitators, can help organizations overcome these human and technical challenges. Facilitators typically have excellent communication and interpersonal skills, and they draw from an arsenal of collaboration techniques to design and execute productive work practices on behalf of the teams they serve. As the group works, the facilitator monitors for and intervenes to improve emerging issues of communication, reasoning, information access, distraction, and goal congruence. Research shows that facilitators, supported by collaboration technology, can reduce a group's project cycle time by as much as 90 percent (Fjermestad and Hiltz 2001). Facilitators, however, can be a costly option for an organization, and so, many groups that could benefit from their services do not have access to them. Further, it can be challenging for an organization to retain its facilitators because, as articulate, problem-solving people-oriented employees who are comfortable with technology, they are often either promoted to new positions or they leave the organization to establish consulting practices (Agres et al. 2005).

Over the past decade, researchers have, therefore, been developing, applying, and evaluating ways to design productive, task-specific work practices that practitioners, who are not professional facilitators, can successfully execute for themselves. This research has addressed collaboration from a holistic perspective: focusing simultaneously on the details of a work practice, the configuration and packaging of required technology, and documentation of the guidance that practitioners must give a group to move it through useful patterns of collaboration toward its goals. This stream of research has come to be called Collaboration Engineering.

Collaboration Engineering concerns the design and deployment of collaboration processes for recurring high-value collaborative tasks. In Collaboration Engineering, a *collaboration engineer* designs a reusable and predictable collaboration process for a recurring task including technological support, and transfers the design to *practitioners* to execute for themselves without the ongoing intervention of *group process professionals*, i.e., facilitators. These practitioners are domain experts, but are not necessarily experts in designing new collaboration processes for themselves or others. They execute the designed collaboration process as part of their regular work.

The Collaboration Engineering field is at the crossroads of many disciplines, among them information systems, computer science, systems engineering, organization science, organizational behavior, education, communication, and social, cognitive, and organizational and industrial psychology. Collaboration Engineering researchers often combine insights from these disciplines to find better ways to design a collaborative work process that stimulates self-sustained use by a growing number of practitioners. The need to combine the insights from these various disciplines makes Collaboration Engineering a fertile research domain.

In this editorial we describe the Collaboration Engineering domain and the foundations of the Collaboration Engineering approach. We then discuss the Collaboration Engineering approach in

more detail, highlighting concepts, tools, techniques, and conventions that have emerged in the field. We conclude by reflecting on the Collaboration Engineering research agenda. For all that has been achieved thus far, Collaboration Engineering is, nonetheless, a very new field, with many more questions than answers.

2. Collaboration Engineering Domain

Collaboration Engineering focuses on *mission-critical* collaborative tasks. A mission-critical task is one that creates substantial value, or that reduces the risk of a substantial loss of value for organizational stakeholders. Collaboration Engineering further focuses on processes for mission-critical tasks that are *recurring* and must be executed *frequently*. Examples of frequently recurring collaboration processes can be found in many sectors, for instance financial services, government/defense, and software development:

- Financial services:
 - Collaborative enterprise risk assessment
 - Collaborative service product development
 - Collaborative Sarbanes-Oxley assessments
 - Marketing focus groups
- Government/Defense:
 - Collaborative crisis response
 - Collaborative situational awareness
 - Collaborative course of action analysis
 - Collaborative document creation and review
- Software development:
 - Collaborative requirements negotiation & specification
 - Collaborative usability testing
 - Collaborative requirements inspections
 - Collaborative code inspections

Collaboration Engineering research focuses on frequently recurring processes rather than ad-hoc processes based on the logic of the Technology Transition Model (TTM) (Briggs et al. 2003) and its successor, the Value Frequency Model (VFM) (Briggs and Murphy in press). Both TTM and VFM predict that individuals are most likely to accept and adopt a change of technology or work practice that brings them substantial value on frequent basis. If improvements are realized for a repeated process, then the organization derives benefit from the improvement again and again. If the focus were on ad-hoc processes, then the value of each process improvement would be obtained only once. In addition, in the case of repeatable processes, practitioners of the process can attain results similar to those of professional facilitators without having to master the complete suite of facilitation skills. They need only learn the small sub-set of techniques necessary to conduct their own work practices.

Various field studies have reported successful implementations of processes designed by collaboration engineers. In these situations, the deployed collaboration processes are conducted by self-sustaining practitioners. (Example 1 presents a summary of one of these cases). A sample of these studies includes the following:

- ING Group, a financial services firm, conducts collaborative Risk & Control Self Assessments processes in all of its branches across the world (Vreede and Briggs 2005). This case situation is described in more detail in Example 1.
- The U.S. Army's Advanced Research Lab uses a repeatable collaborative approach to mission analysis (Harder and Higley 2004; Harder et al. 2005).
- The European Aeronautic Defense and Space company (EADS) deployed a repeatable process for Manufacturing Project Knowledge Elicitation (Graaff et al. 2005).
- The Rotterdam Port Authority in the Netherlands has used engineered work practices to support collaborative crisis response training and operational execution (Appelman and Driel 2005).
- A process for collaborative usability testing was successfully employed for the development of a governmental health emergency management system (Fruhling and Vreede 2005).

- A telecom company used a repeatable collaboration process to define and explore new mobile services (Bragge et al. 2005).
- Dozens of groups engaged in effective collaborative software requirements negotiations using the EasyWinWin process (Boehm et al. 2001; Grünbacher et al. 2005).

Example 1: An Engineered Collaborative Work Practice for ING Group.

Following industry guidelines, ING Group was faced with the challenge to perform regular operational risk assessments. In 2002, the organization's management opted for a collaborative approach, where operational risk managers would work with business unit employees directly to help them identify and assess operational risks and define mitigating controls. The organization, therefore, needed to perform hundreds of operational risk management (ORM) workshops around the globe on an annual basis. Although they knew *what* had to be done in an ORM, they did not know *how* to do it in groups. They requested that a collaboration engineer develop a repeatable collaborative ORM work practice that operational risk managers could learn to execute by themselves. Drawing on the experiences and expertise of ING's ORM domain experts, the collaboration engineer developed the first prototype of a work practice, which was called the Risk & Control Self Assessment (R&CSA) process. This was then evaluated and refined by conducting a series of pilot projects within one business unit. After a number of modifications and revisions to the activities and techniques of the work practice, the collaboration engineer showed the R&CSA approach to a group of 12 senior ORM experts. During a half-day walk through, the wording and order of activities was further modified, and proposed collaborative activities were tested using several different facilitation techniques. Once the work practice had been perfected, the collaboration engineer developed documentation and training materials, and began to offer two-day training workshops on R&CSA to ING personnel. To date, more than 250 ORM practitioners have taken the training. Those practitioners have trained other practitioners, who have, in turn, trained others, giving rise to a self-sustaining and growing community of practice for R&CSA. These practitioners have moderated thousands of R&CSA workshops in the field.

3. Collaboration Engineering Foundations

A central foundation for Collaboration Engineering is the use of design patterns to support the design and transition of collaborative work practices. Design patterns were first proposed by Alexander et al. (p. x, 1977): "*Each pattern describes a problem which occurs over and over again in our environment and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.*" A collection of related design patterns can be codified as a pattern language.

Design patterns and pattern languages serve several purposes (Alexander 1979). They provide a convenient common language for communication. They allow designers who know the pattern language to name and share complex concepts without having to explain them over and over again in detail. Individual design patterns can be combined to design larger systems. Alexander's patterns, for example, can be combined to create houses, towns, and communities. Moreover, Alexander (1980) argues that using a pattern language will result in more coherent systems, rather than loosely-coupled individual components. Finally, patterns support teaching, capturing, and sharing expert design knowledge.

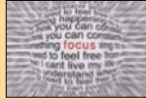
Patterns and pattern based design have found their way into software engineering, (Gamma et al. 1995), workflow management (Aalst et al. 2003), and project management (Khazanchi and Zigurs 2006). For example, Lukosch and Schümmer (2004) propose a pattern language for the development of collaborative software.

The design patterns used in Collaboration Engineering are called thinkLets (Vreede et al. 2006). A thinkLet is a named, scripted technique for predictably and repeatedly invoking known effects among people working together toward a goal. ThinkLets researchers seek to distill each thinkLet to the smallest unit of intellectual capital necessary to predictably invoke a desired effect. ThinkLets are

reusable, transferable facilitation techniques that can be used to move a group through a process toward its agreed goal (Briggs et al. 2003). They enable rapid development of sophisticated, coherent, multi-layered collaboration processes that can improve the productivity and quality of work life for teams (Vreede et al. 2006). Example 2 depicts both sides of cue card for the FastFocus thinkLet. FastFocus is a facilitation technique for moving a group from having many ideas to focusing on fewer that they deem worthy of more attention.

Example 2. Cue card for the FastFocus ThinkLet. Words enclosed in angle brackets (<>) are parameters that are replaced with task-specific terms when the thinkLet is instantiated in a collaboration process design.

FastFocus (Reduce, Clarify)



- Choose this thinkLet...
 - To quickly extract a clean, non-redundant list from a brainstorm activity
 - When agreement on the meaning of the resulting list is important
- Do not choose this thinkLet...
 - To reach consensus on the merits of ideas. Consider the StrawPoll thinkLet instead.
- Setup
 - Distribute all brainstorming ideas across multiple pages – at least one page per participant. Give each participant a page of ideas.
 - Create a place where you can create a publicly viewable list of the key ideas they extract from their brainstorming activity
- Script
 - Say this:
 - “Read through the brainstorming ideas in the page in front of you and look for <important> <ideas>”
 - “I will call on each of you in turn. <Person name>, what is the most <important> <idea> on the page in front of you that is not yet on the list?”
 - Do this:
 - Write each new idea on the public list
 - Say this if someone proposes an idea that may already be on the list :
 - “Is that idea the same or different from <idea X> on the list?”
 - Do This:
 - When all participants have had a turn, tell everybody to swap pages; start a second round
 - For the third round, ask the whole group if anyone has an <important> <Idea> that is not yet on the list
 - Repeat until no one wants to add anything new to the list

Copyright 2007 Briggs and Vreede. Used by written permission.

In Collaboration Engineering, thinkLets are used as building blocks for team process designs in many domains where collaboration is required (Vreede and Briggs 2005). Each time a thinkLet is instantiated in a design, its parameters may differ, but, nonetheless, predictable group dynamics will emerge. For example, Figure 1 depicts a process model of a collaborative risk identification process consisting of a sequence of four thinkLets. This four-thinkLet process is a segment of a larger design that has been adopted by ING Group as described in Example 1.

Field trials with thinkLets confirmed that novice practitioners found it relatively easy to master thinkLets and thinkLet-based process designs. It was often possible for practitioners to successfully lead a thinkLets-based process after one or two days of training, rather than the weeks or months of apprenticeship normally required (Agres et al. 2005; Vreede and Briggs 2005). Moreover, field experience revealed that when two facilitators know the same set of thinkLets, they can transfer sophisticated thinkLet-based collaboration process designs between themselves with no more than a

page or two of documentation (Vreede and Briggs 2005). To date, about 60 thinkLets have been codified. A few examples of thinkLets are summarized in Table 1.

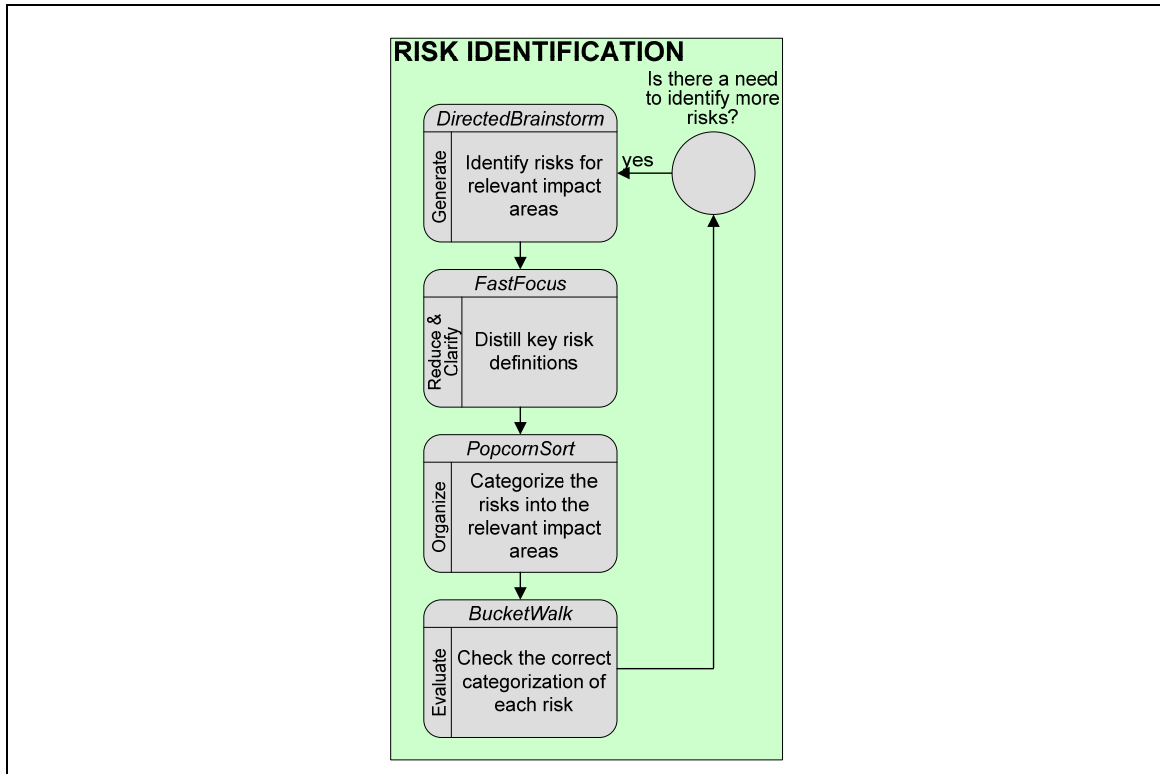


Figure 1. A thinkLets sequence for a risk identification process

Table 1. A list of thinkLets along with their patterns and purposes. Full documentation of a thinkLet requires three to five pages of detail

ThinkLet Name	Pattern of Collaboration	Purpose
DirectedBrainstorm	Generate	To generate a broad, diverse set of highly creative ideas in response to prompts from a moderator and the ideas contributed by team mates.
LeafHopper	Generate	To generate ideas in depth and detail on a focused set of topics.
DealersChoice	Generate	To have different team members generating ideas about different assigned topics in parallel
FastFocus	Reduce & Clarify	To extract a list of key ideas from a raw set of brainstorming comments, and to assure that team members agree on the meaning and phrasing of the items on the resulting list.
FastHarvest	Reduce & Clarify	To have pairs of team members extract a list of key ideas on assigned topics from a raw set of brainstorming comments.
PopcornSort	Organize	To quickly organize a large set of ideas into categories.
StrawPoll	Evaluate	To evaluate a number of concepts with respect to one or more criteria.
MoodRing	Build Commitment	To continuously track the level of consensus within the group with regard to the issue currently under discussion.
CrowBar	Build Commitment	To discover and discuss the reasons behind disagreement on certain issues.

When executed by a group, a thinkLet invokes a rhythm of activities that can be recognized over time as a pattern of collaboration. To date, the following six general patterns of collaboration have been identified (Briggs et al. 2006):

- *Generate*: To move from having fewer concepts to having more concepts in the set of ideas shared by the group. The goal of generation is for a group to gather or create concepts that have not yet been considered by the group. Brainstorming is an example of a generation process.
- *Reduce*: To move from having many concepts to having a focus on fewer concepts deemed worthy of further attention. The goal of reduction is for a group to decrease their cognitive load by limiting the number of concepts they must address. Reduction can be achieved by at least two strategies. The first concerns filtering – eliminating some concepts from consideration. The second concerns abstracting a general concept from multiple specific instances.
- *Clarify*: Moving from less to more shared understanding of the meaning of concepts shared by the group. This is important because people frequently use the same label for different concepts, and use different labels for the same concepts. People on a team also frequently use labels and concepts that are unfamiliar to others on the team.
- *Organize*: To move from less to more understanding of the relationships among the concepts. The goal of organization is to reduce the effort of a follow-on activity. The group might, for example, organize a mixed list of ideas into a number of categories or arrange them into a hierarchical structure.
- *Evaluate*: To move from less to more understanding of the benefit of concepts toward attaining a goal. The goal of evaluation is to focus a discussion or inform a group's choice based on a judgment of the worth of a set of concepts with respect to a set of task-relevant criteria. For example, an evaluation process may involve having a team use a five-point scale to rate the merits of a set of alternatives, or they may conduct a qualitative analysis of the pros and cons of a proposed concept.
- *Build Commitment*¹: To move from having fewer to having more people who are willing to commit to a proposal for moving a group toward its goals. The need to build commitment manifests as individuals contemplate joining, as they build consensus around proposed courses of action, and in the many other decisions that groups make. We define *commitment* as a felt obligation to expend resources and effort to fulfill the terms of an agreement. The goal of commitment building is to let a group of mission-critical stakeholders arrive at mutually acceptable agreements. A group might, for example, seek to build consensus around proposed controls for mitigating key operational risks.

4. Collaboration Engineering Approach

As an approach, Collaboration Engineering consists of a *design phase*, where the repeatable collaboration processes are designed and piloted, and a *deployment phase*, where the new collaboration process is introduced into the organization and practitioners are trained. A high-level overview of the two phases is given in Figure 2.

The design phase starts with the identification and definition of a recurring collaborative task that can benefit from a Collaboration Engineering design effort. A collaboration engineer also identifies best practices for this task. These practices are often found in organizational standards, industry standards, or reference literature. In addition, the collaboration engineer has to gather knowledge on the context in which the collaboration process will be executed. This involves, for example, determining relevant characteristics of the groups executing the process, their stakes involved in the process outcomes, and the required task-relevant competencies of the practitioners that will guide the process execution.

Next, the collaboration engineer uses these insights to create a first version of the collaboration process design (Kofschoten and Vreede in press). In this design effort, the collaborative task is decomposed into a logical sequence of activities that require a pattern of collaboration to be executed by a group. These patterns of collaboration can be created using the collaboration process design patterns, i.e., thinkLets. In other words, the decomposition provides a basis for matching available thinkLets to the constituent activities of the collaborative task, see Figure 3.

¹ In earlier works, this pattern was called, "*Build Consensus*." Subsequent research, however, suggested that consensus building was an instance of the more general concept of building commitment.

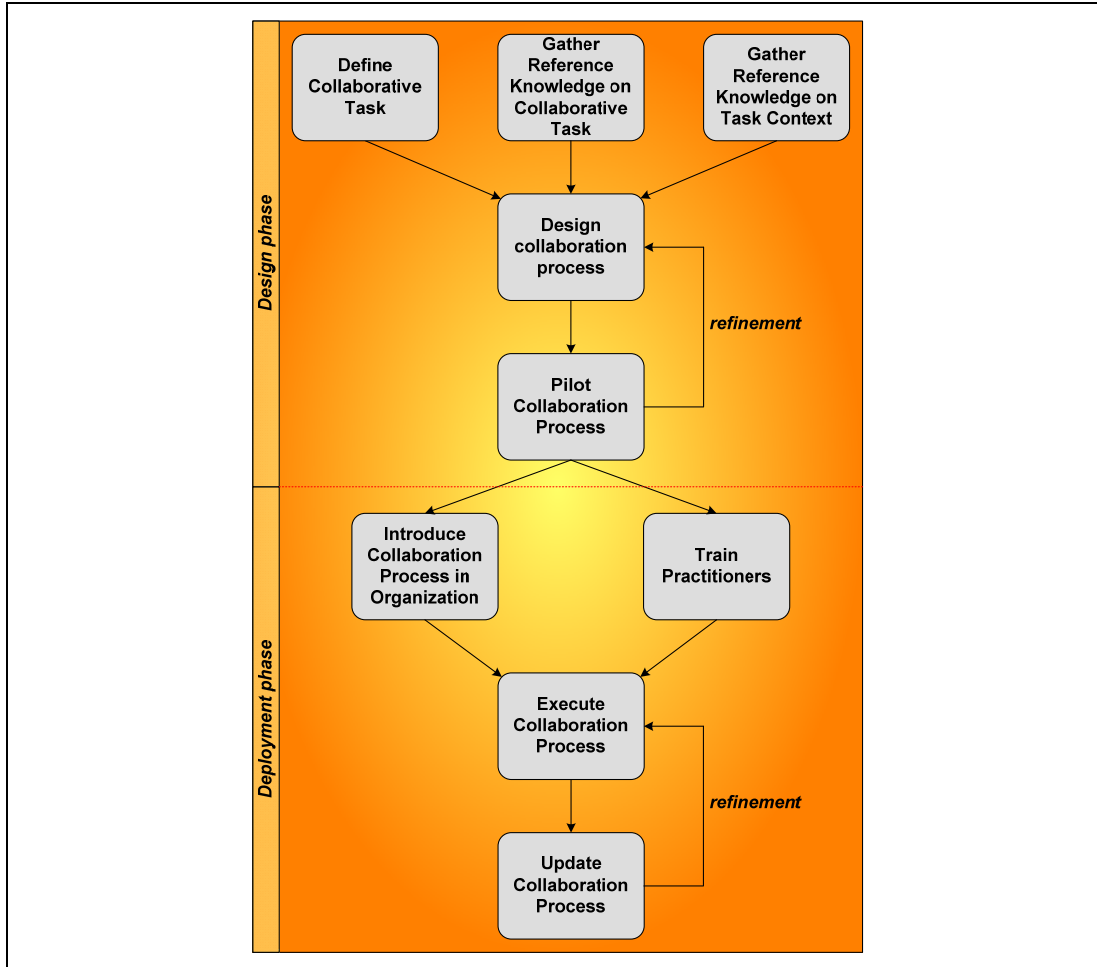


Figure 2. Overview of the Collaboration Engineering approach

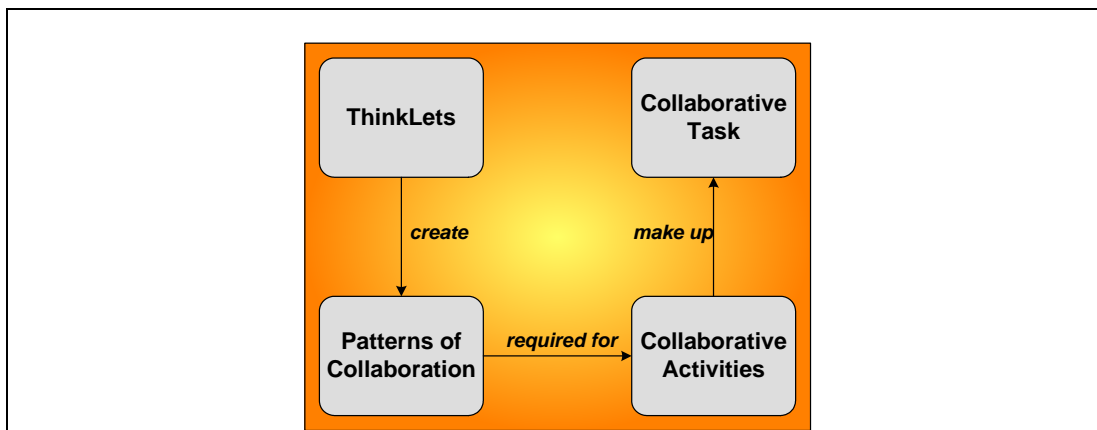


Figure 3. Decomposition of a collaborative task into a sequence of collaboration process design patterns, i.e., thinkLets

Once the first version of the collaboration process design is completed, it can be validated and executed during one or more pilots. The results of these pilots can lead to refinements to the process. If the pilot results are satisfactory, the collaboration process can be implemented in the organization, starting the deployment phase of the Collaboration Engineering approach.

In the deployment phase, the collaboration process is introduced into the organization. This involves, for example, briefing the relevant stakeholders that will be involved in the process and defining a program of incentives for executing the process according to the new standard. Also, practitioners have to be trained to become effective group leaders for the recurring collaboration process. Field research to date has shown that the best results are achieved by combining different training methods, including lectures (for example, on patterns of collaboration, thinkLets, and the role of the practitioner); an exercise for the practitioners to construct the activity flow of the collaboration process themselves; simulation and coaching to practice each step of the process in the context of a case situation; and execution support in the form of thinkLet cue cards and a complete process overview (Kolfchoten et al. in press). Further feedback and experiences with the collaboration process in practice may result in adaptations and improvement. In larger projects, as in the case of ING Group, communities of practice can be formed among the practitioners to exchange experiences and improve or adapt the process to changes in the organization (Chakrapani 2005).

Although the design and deployment activities are described and depicted above in a seemingly linear fashion, it should be noted that in reality, they are not linear in nature. Depending on the context, the Collaboration Engineering approach requires and allows for iteration and incrementation. Certain design activities are carried out in parallel and on different levels of abstraction. For example, an exploration of existing best practices and the design of a process in terms of steps and patterns of collaboration can occur simultaneously. Also, during the piloting of the collaboration process, the collaboration engineer may continuously evaluate the design results so far, together with an organizational counterpart, and make changes accordingly. In other words, the Collaboration Engineering design approach is not meant to be a cookbook. Rather, it should be seen as a set of design steps. Experience shows that the order in which these design steps are executed depends on the type, complexity, and scope of the collaboration task, and the existing amount of insight in the organization's collaborative task.

5. Collaboration Engineering Modeling Techniques

An important aspect of designing and deploying repeatable collaboration processes concerns capturing the design artifacts in a useful format. To this end, Collaboration Engineering researchers have developed various techniques to model and document repeatable collaboration processes. Models of a collaboration process should be expressive, comprehensive, unambiguous, and intuitive. Models not only serve as a vehicle of communication among designers, but are also used to present designs to the organization and to support the training of practitioners that will execute the collaboration process. Below we present three modeling conventions that have been developed and widely used over the past few years: the thinkLet documentation format, the Facilitation Process Model, and the Agenda Design Format.

ThinkLet documentation format

A thinkLet has to capture all information required to create a pattern of collaboration in a predictable, transferable way. To provide consistency and comparability, each thinkLet has to be codified using the same documentation template. This template consists of three components: the identification, the script, and selection guidance (see Vreede et al. 2006 for more details and examples):

- **Identification.** Each thinkLet has a name and picture to represent the specific pattern of group behavior that the thinkLet will create. The name and picture are somewhat "catchy" to make them easier to remember and transfer. An explanation of the metaphor that is represented by the name and picture is provided to strengthen retention of the thinkLet. For example, the LeafHopper thinkLet lets participants brainstorm several topics at the same time. The participants appear to be hopping from topic to topic at will.
- **Script.** The script provides the minimum instructions that a practitioner or facilitator should give to the group in order to create the desired group behavior. The script explains the available capabilities and instructs the group as to what actions they should take and what rules they have to follow. In detail, the script defines the following:
 - *Roles* represent a collection of rules that guide the actions of some set of participants. In some thinkLets, different participants must behave according to different rules. For example,

in some generation thinkLets, there may be two roles: a regular participant and a devil's advocate that challenges the other participants to think more critically.

- *Rules* describe the *actions* that participants must execute using the *capabilities* provided to them under some set of *constraints*. In each thinkLet, individual actions are subject to constraints. For example, in a selection activity, the participants are constrained by the maximum number of items that they can select from a list. The combination of the constrained individual actions over time creates the intended dynamics within the group. Note that small changes in the rules can result in very different interactions among participants. For example, an “add” action guided by a “summarize” constraint gives rise to abstraction, synthesis, and generalization, while an “add” action guided by an “elaborate” constraint gives rise to increasingly detailed exposition of present concepts.
- *Capabilities* define the functionalities that tools must provide to support the thinkLet. For example, the LeafHopper thinkLet mentioned earlier requires the following capabilities: One page per brainstorming topic; participants must be able to read and contribute to each page. Different technologies can be used to afford the capabilities. For example, a LeafHopper can be implemented with flip charts, a white board, or with a GSS.
- *Actions* are activities that the participants must perform during the execution of a thinkLet. These represent basic actions, including add, edit, move, delete, relate, or judge concepts.
- *Parameters* define the information that needs to be specified when the thinkLet is to be executed in a particular context. For example, in a generation thinkLet, a brainstorming question must be defined. In an evaluation thinkLet, the voting criteria must be defined.
- Selection Guidance. When designing a collaboration process, a collaboration engineer has to select the most appropriate thinkLet for each activity in the process. To this end, a collaboration engineer has to understand the effects that different thinkLets will create and which thinkLets work better in certain situations than others. To develop this understanding, the template records thinkLet success stories, “tips and tricks” concerning the thinkLet, and choice guidance in terms of “choose this thinkLet when,” and “don’t choose this thinkLet when.”

Facilitation Process Model

A Facilitation Process Model (FPM) is used to display the flow and logical interdependencies between the activities in a collaboration process. An FPM focuses attention on the logic of the flow of the process from activity to activity. An FPM uses three symbols (see Figure 4) to model the flow of a process. Each *activity* in a process is represented by a rectangle with rounded corners that has been divided into five fields. The left upper field indicates the sequence number of the activity. The largest field contains a descriptive name for the activity that conveys what the team is supposed to do. The field on the left names the primary pattern of collaboration to be created during the activity. The thinkLet name to be used for this purpose appears across the top. The upper right corner displays the time required to complete the activity. Each *decision* that may affect the process flow is represented by a circle. Underneath each decision the decision criteria are indicated. Finally, each *flow direction* is represented by an arrow. Underneath or next to the arrow, the results from a previous activity can be described. These also represent the input for the next activity. Figure 5 depicts a complete FPM for the example in Figure 1.

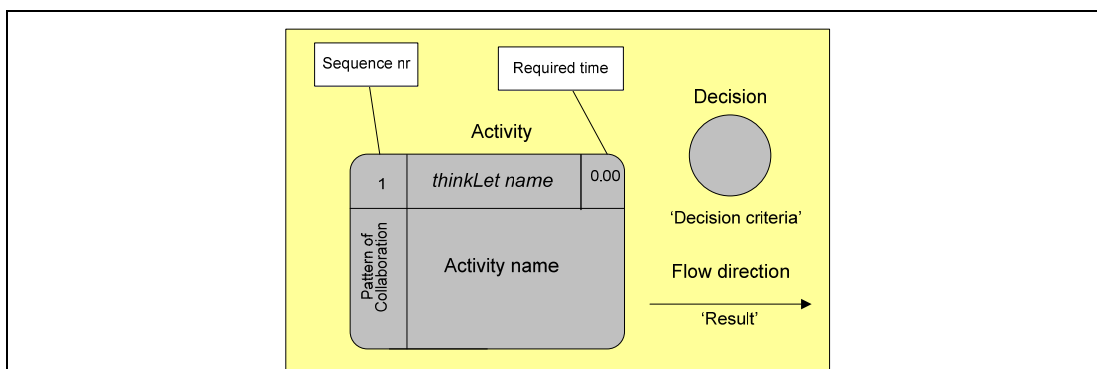


Figure 4. Elements of a Facilitation Process Model

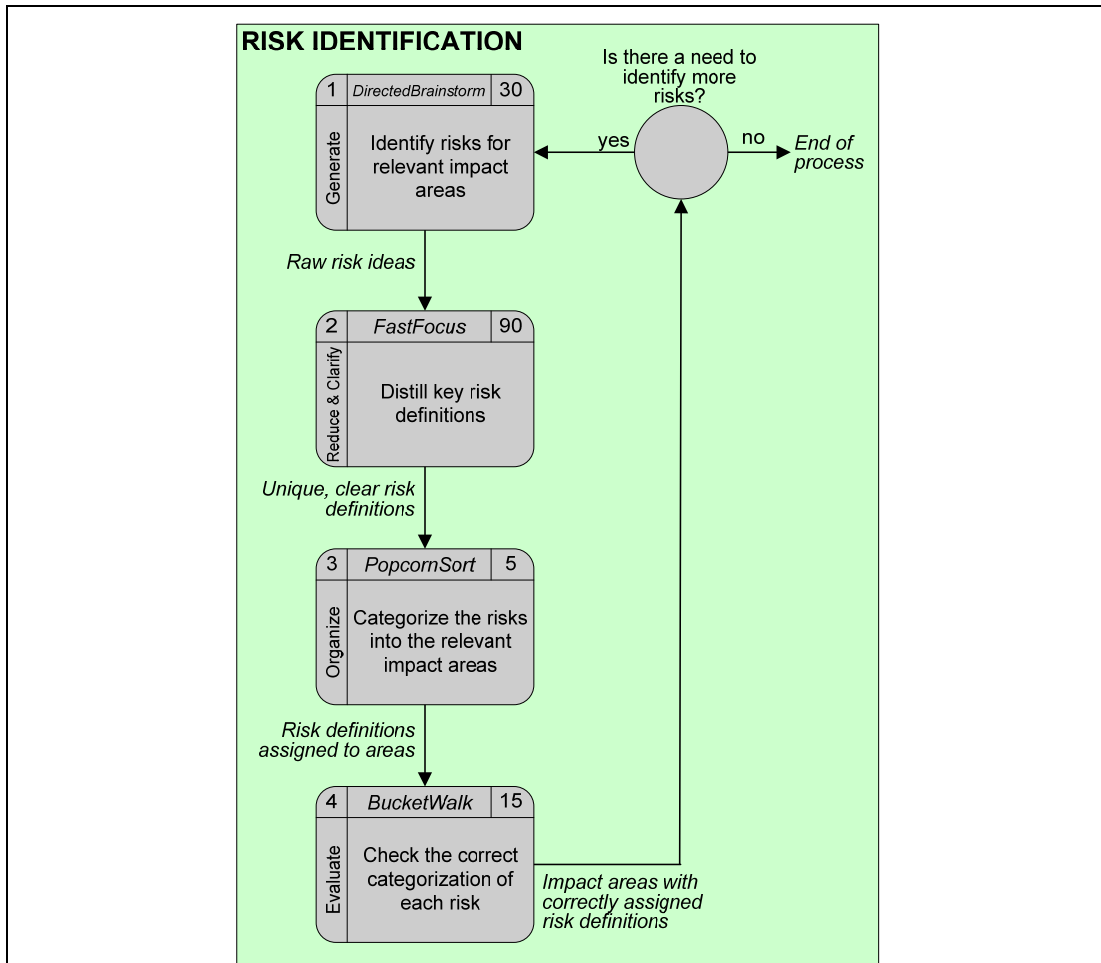


Figure 5. Example of a Facilitation Process Model

Agenda Design Format

To execute a collaboration process design in practice, more information needs to be recorded than the FPM can provide. The Agenda Design Format (ADF) specifies all relevant information for each activity in the process. This information consists of the name of each activity, the specific questions or assignments that will be provided to the group, the deliverables that have to be created in the activity, the thinkLet to be used with the associated pattern of collaboration and tool on which the thinkLet is to be implemented, and finally the starting time of each activity. Table 2 shows the ADF for the risk identification example in Figures 1 and 5, using the GSS Group Systems as a tool platform.

Table 2. Example of the agenda format of a collaboration process design

	Activity	Question/Assignment	Deliverable	ThinkLet (Pattern) Tool	Time
	Introduction to workshop	Introduce goal and deliverable. Goal: Identify key risks for relevant impact areas. Deliverable: A list of clear, unique risk definitions per impact area	Commitment to the goal, understanding GSS, knowing each other	None	9.00
1	Identify risks for relevant impact areas	What are the key risks for the following impact areas: front office, back office, IT, product development, management?	Broad collection of raw risk ideas for various impact areas	DirectedBrainstorm (Generate) EBS	9.20
2	Distill key risk definitions	Please identify and reformulate the most important risk on your sheet	List of unique and clearly defined risks	FastFocus (Reduce & Clarify) EBS and Categorizer	9.50
3	Categorize risks into relevant impact areas	Please place each risk definition into the impact area that is responsible to manage it.	Initial distribution of risks over responsible impact areas	PopcornSort (Organize) Categorizer	11.20
4	Check correct categorization of each risk	Please check for each impact area whether all risks in there have been properly assigned.	Agreed on assignment of risks over responsible impact areas	BucketWalk (Evaluate) Categorizer	11.25
	Decide on whether to identify more risks	If there are sufficient risks defined for each area, then conclude workshop, else go back to step 1 for impact area(s) concerned	Decision on whether to identify more risks	None	11.50

6. Collaboration Engineering Research Agenda

Since the start of Collaboration Engineering research in 2001, more than a 100 scholarly works have been published by researchers across the world. Many field applications have taken place. Research efforts have focused on various theories underlying Collaboration Engineering and on the development of metrics and instruments to assess the quality of Collaboration Engineering interventions and designs (see Figure 6). Studies have employed a variety of research strategies, including field studies like case studies and action research, laboratory experiments, and prototype development. Studies have taken place in different physical environments (e.g., face to face or virtual collaboration) and in different or mixed social or cultural settings.

Although many encouraging results have been reported, many academic and practical challenges and opportunities lie ahead to further develop the Collaboration Engineering research area in terms of its foundations, its design and deployment approach, and its modeling techniques. Below, we sketch a number of these research challenges and opportunities.

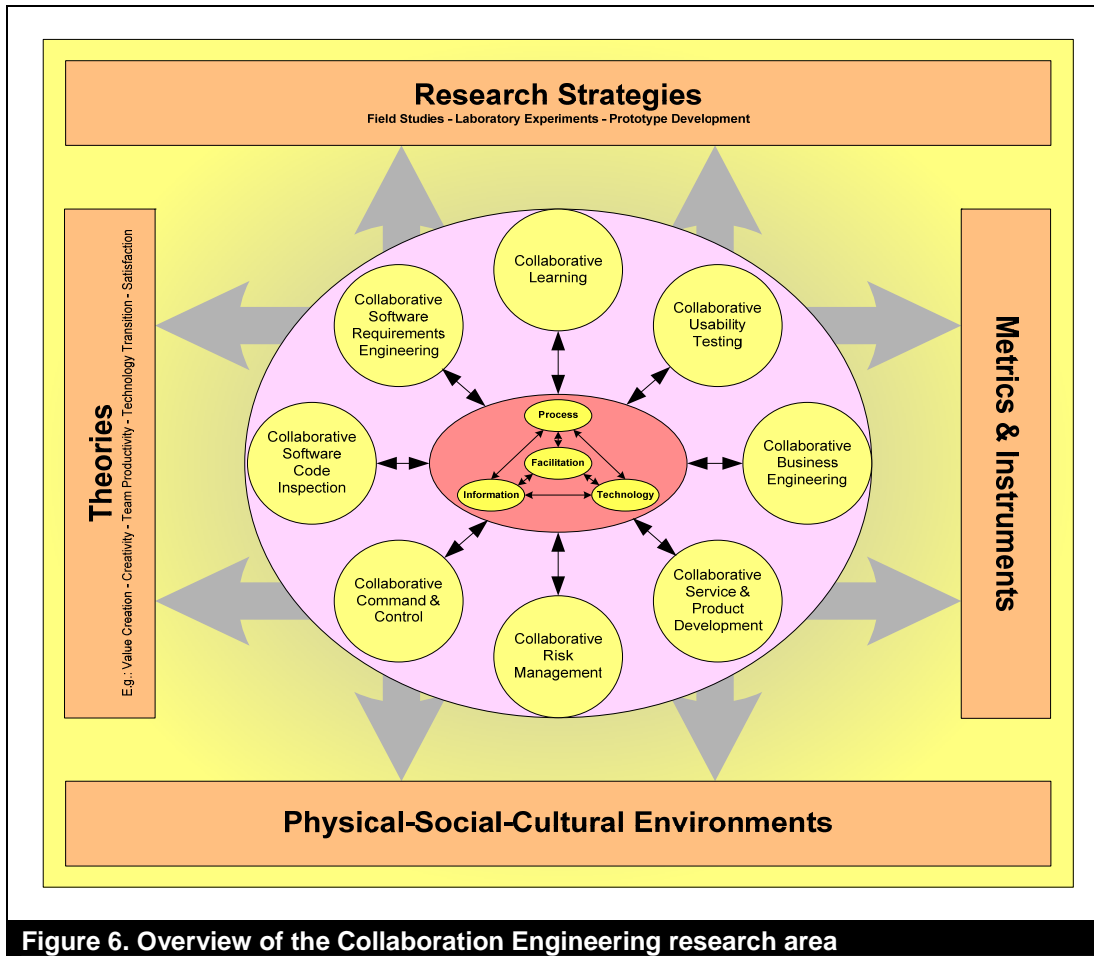


Figure 6. Overview of the Collaboration Engineering research area

Foundations

There is a plethora of research opportunities with respect to the foundations of Collaboration Engineering. On the group level, decades of research have yielded many insights into group behavior in the context of a particular group task, see e.g. (Dennis 2001; Fjermestad and Hiltz 1999) for overviews. The literature shows that much of this research has focused on brainstorming. Deeper theoretical understanding of other collaborative activities is greatly needed. In particular, future research should focus on the theoretical foundations of the reduce, clarify, evaluate, organize, and build consensus patterns of collaboration.

On the organizational level, there is a need for further fundamental research on how groups and organizations accept, adopt, and adapt repeatable collaboration processes. How do groups embrace a standard repeatable collaboration process over time? How do they change it themselves over time to better suit their needs?

Another fundamental challenge concerns the quality assessment of a collaboration process design. How can we measure the design of a collaboration process either before it is executed (i.e. the "paper" design) or during execution? And, is it possible to create a quality assessment framework that is independent from the specific collaboration process that is being assessed or its context?

Collaboration Engineering has focused mainly on designing and deploying 'fixed' collaboration process design: a standard sequence of collaborative activities modeled with collaboration design patterns (thinkLets). However, there are situations in which a single standard sequence cannot offer adequate support. These are situations where more creative, ad hoc solutions have to be found for recurring collaborative challenge, for example, in crisis response situations. Thus, it may be possible

to define a standard repository of a limited number of thinkLets that a group can use to create adaptive process sequences as and when it needs them.

More research is also required to explore whether Collaboration Engineering can only be applied to design organization-specific collaboration processes, or whether it can also be used to design processes that are industry-specific, for example, an industry standard on collaborative software engineering project post-mortems.

Finally, a key issue for collaboration engineers concerns the cultural context in which the repeatable collaboration process has to be executed. To what extent is Collaboration Engineering culturally bound? To what extent can thinkLets-based processes be applied in different cultures? How is the role of a practitioner perceived in different cultures? ING Group's experiences show that their standard collaborative risk assessment process was successfully accepted and applied in more than 30 countries in North and South America, Europe, Asia, and Australia. However, some Asian cultures showed a high reluctance and very low diffusion rates.

Design and Deployment Approach

To date, most Collaboration Engineering studies have focused on face-to-face settings. As virtual team work and virtual work environments are becoming more dominant settings for (inter-)organizational work, the applicability of Collaboration Engineering concepts to virtual collaboration has to be explored. This includes seeking out and capturing effective design patterns and deriving design guidelines for virtual collaboration processes.

Another research opportunity concerns the characteristics of the individuals that fulfill the practitioners' role. Practitioners are domain experts but not collaboration experts. Not everyone who has a deep understanding of an application domain is necessarily suited to be an effective group leader for repeatable processes in this domain. It could increase the likelihood that practitioners become effective group leaders if we have a deeper understanding of the personality characteristics that are shared among successful facilitators. Thus, the question to ask is whether people with a natural flair for guiding group work share similar personalities? If so, then these personality characteristics could be used as a way to identify and select candidate practitioners within an organization.

Modeling Techniques

The current Collaboration Engineering modeling techniques that are used to document collaboration process designs were developed through experiences in a large number of field studies. A next phase in the development of these techniques should focus on strengthening their theoretical basis in two ways. First, the different models should be unified by developing a meta-model that specifies all relevant elements in a collaboration process design and the interdependencies among these elements. Based on that unified meta-model, different aspect models could be formally defined. Each aspect model could highlight a particular perspective on the collaboration process design, just as the FPM is currently highlighting the flow of the process logic and the ADF focuses on the specific instructions given to the group and the desired deliverables in each activity in the process.

Second, based on a unified meta-model, a formal model syntax of each of the modeling techniques could be derived. Such a syntax would provide a basis to ensure that models adhere to a minimum quality standard. They could also provide a starting point to develop guidelines or model checks to (automatically) assess the quality of collaboration process models.

Tool support

There are various opportunities to develop Computer Assisted Collaboration Engineering (CACE) tools. For example, tools can be developed to support design activities. Examples of tool support in this area include, but are not limited to, providing guidance in the choice of thinkLets to match process activities, drawing Facilitation Process Models, or providing automatic design guidance during the construction of an Agenda Design Format.

Tools can also be developed to support the documentation of thinkLets. As thinkLets are used by various collaboration engineers and many experiences are gathered in the field, updates to these

thinkLets are inevitable. To enable consistency and accuracy in the formal definition of each thinkLet according to the template presented in Section 5, a thinkLet content management tool would be useful.

A final area where tool support can advance the Collaboration Engineering area concerns the actual execution of collaboration process designs. Currently, collaboration process design has to be implemented on general collaboration software platforms, such as commercial GSS. As these platforms offer many more functionalities or configurations than are needed in any particular recurring collaboration process, they are very complex to operate for practitioners. To overcome this challenge, a design studio could be developed that allows a collaboration engineer not only to capture the logic of collaboration processes but also the guidance that the practitioner needs to execute it. The studio would then instantiate the design, including guidance as a stand-alone application that the practitioner and his or her group can run any time the process needs to be executed. Such a studio would make collaboration technologies more accessible and useable than general purpose GSS suites.

7. Conclusions

Collaboration is a critical phenomenon in organizational life. Collaboration is necessary yet difficult to do well. The field of IS has devoted much effort to understanding how groups can and will use technologies to improve the productivity of their collaborative work. Over the past decade, the field of Collaboration Engineering has emerged as a focal point for research on designing and deploying collaboration processes that are recurring in nature and that are executed by practitioners in organizations rather than collaboration professionals.

In this editorial we have highlighted the foundations of the Collaboration Engineering field, given an overview of the design and deployment activities and modeling techniques, and sketched a research agenda for the coming decade. The insights presented in this paper represent the results of many studies and field applications that have been made possible through the efforts of an international community of researchers. We hope that past results and future opportunities will inspire many more to become active in this exciting field of research.

Acknowledgments

We are grateful for the many researchers that have become part of a growing community of Collaboration Engineering research that have pushed the research agenda forward. We are particularly indebted to Gwendolyn Kofschoten and Douglas Dean for their deep insights shared in numerous discussions that helped shape the field of Collaboration Engineering to be what it is today.

References

- Aalst, W.M.P van der; Hofstede, A.H.M. ter and Kiepuszewski, B. (2003). Workflow Patterns, *Distributed and Parallel Databases* 14: 5-51.
- Agres, A.; Vreede, G.J. de and Briggs, R.O. (2005). A Tale of Two Cities: Case Studies of GSS Transition in Two Organizations, *Group Decision and Negotiation* 14,(4): 256-266.
- Alexander, C. (1979). *The Timeless Way of Building*, New York, Oxford University Press.
- Alexander, C. (1980) *The nature of Order: an essay on the art of building and the nature of the universe*, The center for environmental structure, Berkeley.
- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S. (1977) *A Pattern Language, Towns, Buildings, Construction*, Oxford University Press, New York.
- Appelman, J.H., Driel, J. van (2005), Crisis-response in the Port of Rotterdam: Can we do without a facilitator in distributed settings?, *Proceedings of the 38th Hawaii International Conference on System Sciences*, IEEE Computer Society Press.
- Boehm, B., Grünbacher, P., Briggs, R.O. (2001), Developing Groupware for Requirements Negotiation: Lessons Learned, *IEEE Software*, May/June, 46-55.
- Bragge, J.H., Hengst, M. den, Tuunanen, T., Virtanen, V. (2005), A Repeatable Collaboration Process for Developing a Road Map for Emerging New Technology Business: Case Mobile Marketing, *Proceedings of AMCIS 2005*, Omaha, NE.
- Briggs, R.O., Murphy, J.D., (in press), Discovering and Evaluating Collaboration Engineering Opportunities: An Interview Protocol Based on the Value Frequency Model, *Group Decision and Negotiation*.

- Briggs, R.O.; Kolfshoten, G.L.; Vreede, G.J. de and Dean, D.L. (2006). Defining Key Concepts for Collaboration Engineering, *Americas Conference on Information Systems*, Acapulco, Mexico, AIS.
- Briggs, R.O.; Vreede, G.J. de and Nunamaker, J.F. Jr (2003). Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems, *Journal of Management Information Systems* 19,(4): 31-63.
- Chakrapani, A. (2005), A Design & Evaluation Framework for Setting Up a Community of Practice, *Master's Thesis*, University of Nebraska at Omaha.
- Dennis, A.R.; Wixom, B.H. and Vandenberg, R.J. (2001). Understanding Fit and Appropriation Effects in Group Support Systems Via Meta-Analysis, *Management Information Systems Quarterly* 25, (2): 167-183.
- Fjermestad, J. and Hiltz, S.R. (1999). An Assessment of Group Support Systems Experimental Research: Methodology and Results, *Journal of Management Information Systems* 15, (3): 7-149.
- Fjermestad, J. and Hiltz, S.R. (2001). A Descriptive Evaluation of Group Support Systems Case and Field Studies, *Journal of Management Information Systems* 17, (3): 115-159.
- Fruhling, A., Vreede, G.J. de (2005), Collaborative Usability Testing to Facilitate Stakeholder Involvement, in: S. Biffli, A. Aurum, B. Boehm, H. Erdogmus, P. Grünbacher (eds), *Value Based Software Engineering*, Berlin: Springer-Verlag, 201-223.
- Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995) *Elements of Reusable Object-Oriented Software*, Addison-Wesley Publishing Company.
- Graaff, J. de, Appelman, J.H., Verburg, R.M., Jordan, R. (2005), Group supported knowledge elicitation: where Knowledge Management meets Groups Support Systems, *Proceedings of Group Decision & Negotiation 2005*, Vienna, Austria, July 10-13.
- Grünbacher, P., Köszegi, S., Biffli, S. (2005), Stakeholder Value Proposition Elicitation and Reconciliation, in: S. Biffli, A. Aurum, B. Boehm, H. Erdogmus, P. Grünbacher (eds), *Value Based Software Engineering*, Berlin: Springer-Verlag, 133-154.
- Harder, R.J., Higley, H. (2004), Application of ThinkLets to Team Cognitive Task Analysis, *Proceedings of the 37th Hawaii International Conference On System Sciences*, IEEE Computer Society Press.
- Harder, R.J., Keeter, J.M., Woodcock, B.W., Ferguson, J.W., Wills, F.W. (2005), Insights in Implementing Collaboration Engineering, *Proceedings of the 38th Hawaiian International Conference on System Sciences*, Los Alamitos: IEEE Computer Society Press.
- Khazanchi, D. and Zigurs, I. (2006). Patterns for effective management of virtual projects: Theory and evidence, *International Journal of e-Collaboration* 2,(3): 25-48.
- Kolfshoten, G.L., Vreede, G.J. de, (in press), A Design Approach for Collaboration Engineering: A Multi-Method Design Science Study in Collaboration Engineering, *Journal of Management Information Systems*.
- Kolfshoten, G.L.; Vreede, G.J. de and Pietron, L. (in press). A training approach for the transition of repeatable collaboration processes to practitioners, *Group Decision and Negotiation*.
- Lukosch, S. and Schümmer, T. (2004) Communicating design knowledge with groupware technology patterns, In *CRIWG 2004*, Vol. LNCS 3198 (Ed, Vreede, G.J. de, Guerrero, L.A., Raventos, G.M., (eds.)) Springer-Verlag, San Carlos, Costa Rica, pp. 223-237.
- Vreede, G.J. de, Briggs, R.O. (2005), Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks, *Proceedings of the 38th Hawaiian International Conference on System Sciences*, Los Alamitos: IEEE Computer Society Press.
- Vreede, G.J. de, Kolfshoten, G.L., Briggs, R.O. (2006), ThinkLets: A Collaboration Engineering Pattern Language, *International Journal of Computer Applications and Technology*.

About the Authors

Gert-Jan de Vreede is a Kayser Distinguished Professor and the Director of the Center for Collaboration Science at the University of Nebraska at Omaha. He is also affiliated with the Faculty of Technology, Policy and Management of Delft University of Technology in the Netherlands from where he received his PhD. He has been a visiting professor at the University of Arizona and the University of Pretoria. His research focuses on field applications of collaboration technologies, the theoretical foundations of collaboration, Collaboration Engineering, and the facilitation of group work. His research has been published in various journals, including *Journal of Management Information Systems*, *Journal of the AIS*, *Communications of the AIS*, *Small Group Research*, *Communications of the ACM*, *DataBase*, *Group Decision and Negotiation*, *International Journal of e-Collaboration*, *Journal of Decision Systems*, *Journal of Creativity and Innovation Management*, *Simulation & Gaming*, *Simulation*, and *Journal of Simulation Practice and Theory*.

Robert O. Briggs is Director of Academic Affairs for the Center for Collaboration Science at the University of Nebraska at Omaha and a professor in the College of Business Administration in the same university. He is a Fellow of the Center for Distance Education in the College of Rural and Community Development at University of Alaska Fairbanks. He researches the cognitive foundations of collaboration and learning and applies his findings to the development and deployment of collaboration technologies and new work practices for collaboration in the workplace. He is co-founder of the emerging research area of Collaboration Engineering and co-inventor of the thinkLets concept. He has published more than 100 peer-reviewed scholarly works. He earned his Ph.D. from University of Arizona in 1994.

Anne P. Massey is Dean's Research Professor and Professor of Information Systems in the Kelley School of Business at Indiana University. She currently teaches graduate courses in innovation and collaboration, and management of IT. Her research interests focus on technology-based innovation and knowledge-intensive processes, including enabling internal and external collaboration using ICTs, social media and virtual worlds. Her research has garnered industry-based support and funding and field work with several companies. Her research has been published in leading journals including *MIS Quarterly*, *Academy of Management Journal*, *Decision Sciences*, *European Journal of Information Systems*, *IEEE Transactions on Engineering Management*, and the *Journal of Management Information Systems*, among others.

Copyright © 2009, by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers for commercial use, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints, or via e-mail from ais@gsu.edu.



Editor
Kalle Lyytinen
 Case Western Reserve University, USA

Senior Editors			
Robert Fichman	Boston College, USA	Dennis Galletta	University of Pittsburgh, USA
Varun Grover	Clemson University, USA	Rudy Hirschheim	Louisiana State University, USA
Robert Kauffman	University of Minnesota, USA	Frank Land	London School of Economics, UK
Jeffrey Parsons	Memorial University of Newfoundland, Canada	Suzanne Rivard	Ecole des Hautes Etudes Commerciales, Canada
Ananth Srinivasan	University of Auckland, New Zealand	Bernard C.Y. Tan	National University of Singapore, Singapore
Michael Wade	York University, Canada	Ping Zhang	Syracuse University, USA
Editorial Board			
Steve Alter	University of San Francisco, USA	Kemal Altinkemer	Purdue University, USA
Michael Barrett	University of Cambridge, UK	Cynthia Beath	University of Texas at Austin, USA
Michel Benaroch	University of Syracuse, USA	Francois Bodart	University of Namur, Belgium
Marie-Claude Boudreau	University of Georgia, USA	Susan A. Brown	University of Arizona, USA
Tung Bui	University of Hawaii, USA	Andrew Burton-Jones	University of British Columbia, Canada
Dave Chatterjee	University of Georgia, USA	Patrick Y.K. Chau	University of Hong Kong, China
Mike Chiasson	Lancaster University, UK	Mary J. Culnan	Bentley College, USA
Jan Damsgaard	Copenhagen Business School, Denmark	Samer Faraj	McGill university, Canada
Chris Forman	Carnegie Mellon University, USA	Ola Henfridsson	Viktoria Institute & Halmstad University, Sweden
Hitotora Higashikuni	Tokyo University of Science, Japan	Kai Lung Hui	National University of Singapore, Singapore
Hemant Jain	University of Wisconsin-Milwaukee, USA	Bill Kettinger	University of South Carolina, USA
Rajiv Kohli	College of William and Mary, USA	Mary Lacity	University of Missouri-St. Louis, USA
Ho Geun Lee	Yonsei University, Korea	Jae-Nam Lee	Korea University
Kai H. Lim	City University of Hong Kong, Hong Kong	Ji-Ye Mao	Renmin University, China
Anne Massey	Indiana University, USA	Emmanuel Monod	Dauphine University, France
Michael Myers	University of Auckland, New Zealand	Fiona Fui-Hoon Nah	University of Nebraska-Lincoln, USA
Mike Newman	University of Manchester, UK	Jonathan Palmer	College of William and Mary, USA
Paul Palou	University of California, Riverside, USA	Brian Pentland	Michigan State University, USA
Yves Pigneur	HEC, Lausanne, Switzerland	Jaana Porra	University of Houston, USA
Sandeep Purao	Penn State University, USA	T. S. Raghu	Arizona State University, USA
Dewan Rajiv	University of Rochester, USA	Balasubramaniam Ramesh	Georgia State University, USA
Timo Saarinen	Helsinki School of Economics, Finland	Susan Scott	The London School of Economics and Political Science, UK
Ben Shao	Arizona State University, USA	Olivia Sheng	University of Utah, USA
Carsten Sorensen	The London School of Economics and Political Science, UK	Katherine Stewart	University of Maryland, USA
Mani Subramani	University of Minnesota, USA	Burt Swanson	University of California at Los Angeles, USA
Dov Te'eni	Tel Aviv University, Israel	Jason Thatcher	Clemson University, USA
Ron Thompson	Wake Forest University, USA	Christian Wagner	City University of Hong Kong, Hong Kong
Eric Walden	Texas Tech University, USA	Eric Wang	National Central University, Taiwan
Jonathan Wareham	ESADE, Spain	Stephanie Watts	Boston University, USA
Bruce Weber	London Business School, UK	Tim Weitzel	Bamberg University, Germany
Richard Welke	Georgia State University, USA	George Westerman	Massachusetts Institute of Technology, USA
Kevin Zhu	University of California at Irvine, USA	Ilze Zigurs	University of Nebraska at Omaha, USA
Administrator			
Eph McLean	AIS, Executive Director	Georgia State University, USA	
J. Peter Tinsley	Deputy Executive Director	Association for Information Systems, USA	
Reagan Ramsower	Publisher	Baylor University	