A Transfer Approach for Facilitation Knowledge in Computer-Supported Collaboration

Stefan Werner Knoll[1], Jana Schumann[2], Thomas Matzdorf[2], Ayneta Adege[2], Martin Linnemann[2], and Graham Horton[2]

[1] Delft University of Technology, Faculty of Technology Policy and Management,
Systems Engineering Department, Jaalaan 5, 2628BX Delft, The Netherlands
`s.w.knoll@tudelft.nl`
[2] University of Magdeburg, Faculty of Computer Science, Department of Simulation and Graphics, Universitaetsplatz 2, 39106 Magdeburg, Germany
`{jana.schumann,thomas.matzdorf,ayneta.adege,`
`martin.linnemann}@st.ovgu.de,`
`graham@sim-md.de`

Abstract. Collaboration is an important process for companies to combine the potential and expertise of their employees. Groupware can improve the productivity of collaboration by coordinating activities and improving group communication. Considering the possible complexity of a collaboration process, the faithful appropriation of a groupware technology is fundamental to design predictable and efficient collaboration. This paper presents ongoing research on how to improve technological support for collaboration by formalizing the workflow of a collaboration process into a machine-readable process description. We will present a knowledge transfer approach for the adaptation of a logical process design by an inexperienced user. This approach transfers facilitation knowledge for the selection and configuration of a collaboration process and provides rules for instructional writing to support an inexperienced user in defining clear and explicit instructions. A software application was used to evaluate the knowledge transfer approach in a quasi-experiment with inexperienced participants.

Keywords: Groupware, Collaboration, Facilitation, Instructional Design, Knowledge Transfer Approach.

1 Introduction

By definition, collaboration describes a group process where participants work together to achieve a shared goal [34]. Over the years, the research focus on collaboration has changed from groups whose members work in the same place to geographically distributed virtual groups. This results from the fact that virtual groups using temporary technological support for collaboration comprise an important structural component of many multinational organizations [29]. The collaboration process and its outcomes are affected by different internal and external factors like the

A.S. Vivacqua, C. Gutwin, and M.R.S. Borges (Eds.): CRIWG 2011, LNCS 6969, pp. 110–125, 2011.

characteristics of the individuals, the task, the context, and the technology used [8], [28]. Different theories exist that describe and predict the influence of these factors on group behaviors and performances in relation to group communication [31], group participation [7], [12], [16] and group cohesiveness [13], [15]. From the literature, our conclusion is that most of the influencing factors cannot be generalized for collaboration in general. Depending on the given process characteristics, shared rules, norms and structures can be necessary to handle negative group behaviors and support group performance.

Collaboration support can consist of tools, processes and services that support groups during the design and execution of collaboration. Technological support for collaboration is given by groupware technologies, which offer a variety of local and web-based applications to structure collaborative activities and improve group communication [10], [28]. Today, a huge amount of web-based applications exist that provide different functionalities and mechanisms [26]. Research indicates that the faithful appropriation of a groupware technology is fundamental to design predictable and efficient collaboration [9], [11]. However, the adaptation of a groupware technology to a designed collaborative process can lead to a high conceptual load for inexperienced users [4]. The faithful appropriation of a groupware technology in virtual groups is still a challenge in collaboration support.

Our research analyzes the use of the Collaboration Engineering approach [4] to improve technological support for collaboration by formalizing the workflow of a collaboration process into a machine-readable process description. A generic groupware technology could use the underlying process logic of the workflow to provide functionalities and knowledge that support inexperienced users during the design and execution of a collaboration process. In earlier work, we developed a modeling language for collaboration, which illustrates different pieces of process information like process activities, process data and process events to define the workflow of a collaboration process [18].

A prototype of a groupware technology was designed that uses this process information to guide a group automatically through a process and to provide interfaces and facilitation instructions that allow the group to execute the defined collaborative activities [19]. By using the modeling language, different logical process designs were developed which could be adapted to different tasks by changing parameters and facilitation instructions. However, this adaptation of a logical design can be difficult for users with less expertise in facilitation, who cannot determine the effect of their change on the effectively of collaboration process.

Different studies have analyzed the influence of facilitation on collaboration using technological support [24], [25], [36]. They indicate that expertise of a facilitator is tacit knowledge, which is difficult to transfer to a group of inexperienced users [20]. To capture and share knowledge about collaboration, different approaches can be found such as handbooks for group facilitation [32] or the use of a pattern approach in a technological environment, technological support for agenda building [30] or the computer aided collaboration engineering tool [22].

In this paper, we present a knowledge transfer approach for the adaptation of a logical process design by an inexperienced user. This approach is based on a pattern approach and transfers facilitation knowledge for the selection and configuration of a collaboration process and provides rules for instructional writing to support an inexperienced user in defining clear and explicit instructions. We combined the knowledge transfer approach with our approach of a modeling language for collaboration and developed a software application to support the user in adapting a logical process design.

In the following sections, we first provide a short introduction to our research and related work. Then we present our approach as well as its implementation into a software application. We further present the setup and results of our evaluation experiment, before we conclude with a summary and outlook on future work directions.

2 Background

Briggs et al. [4] assume that the expertise needed for design and execution of collaboration can be reduced by packing and transferring knowledge about collaboration. They introduce Collaboration Engineering as a facilitation, design and training approach for collaboration work practices that can be executed without ongoing support from collaboration professionals such as facilitators. To reach this goal, Collaboration Engineering classifies collaboration into six key patterns of collaboration: Generate, Reduce, Clarify, Organize, Evaluate and Build Consensus [5]. Each pattern stands for different reusable collaborative activities of a group that can be used over a period of time to move from a defined starting state to an intended end state [21]. The concept of the thinkLet was introduced as a design pattern to collect, create, document and test these collaborative activity of a group [5], [35]. Each thinkLet provides information for its selection and how to create a required pattern of collaboration by using a technology in a defined configuration. According to the given design approach for collaboration processes [23], a collaboration process can be decomposed into a sequence of design pattern thinkLets. Each thinkLet contains knowledge about the used setting and its configuration for a given task as well as a script for each activity in the process that is needed to engender the pattern of collaboration. A designed collaboration process is documented as a paper-based handbook. Tacit knowledge and skills for the use of the handbook will be transferred in a training approach [23]. Research indicates that groups who are trained in using thinkLets can predictably and repeatably engender the pattern of collaboration that a given thinkLet is intended for, even without any facilitation expertise [35].

We believe that Collaboration Engineering represents an interesting approach to support collaboration using technological support. The concept of thinkLets as a design pattern can be used to transfer knowledge about the configuration and use of a specific groupware technology. Currently, a collaboration process design will be documented as a paper-based handbook. To support the faithful appropriation of a specific groupware technology, the handbook needs to be closely connected to the used technology, which reduces the transferability of a collaboration process design for other groupware technologies.

2.1 A Modeling Language for Collaboration

In earlier work, we analyzed the applicability of the Collaboration Engineering approach to logical process descriptions similar to the concept workflow of Business Process Engineering [14]. The resulting logical model for collaboration illustrates different pieces of process information like process activities, process data and process events to define the workflow of a collaboration process [18]. The modelling language adopts the design pattern thinkLet as a process template that creates one known collaboration pattern. ThinkLets can be combined to different collaboration processes, which can be adapted to a group goal by the configuration of their parameters and activities. The thinkLet script defines a sequence of abstract actions a group of participant must do to achieve an intended collaboration pattern. However, these actions only provide abstract guidelines for facilitation, details facilitation skills must be transferred by a training approach.

Research indicates that the quality of facilitation is vital for collaboration success [27], [36]. As a result, we believe that a rule of a thinkLet script should include information about facilitation as a formal instruction entity. Based on the Shannon-Weaver Model for communication [33], we introduced a design approach for a reusable formal instruction element called thinXel, which represents an instance of a thinkLet rule. The concept thinXel is originally defined as an atomic facilitator instruction, leading to a response of the participants that has a well-defined function in the context of the group goal [17]. Experimental results have shown that by using thinXels during collaboration, misunderstanding of facilitation instructions by the participants can be reduced [17]. Furthermore, the attention of the participants can kept on the collaboration process.

Currently our modeling language for collaboration can be expressed in a graphical or a semantical process notation. The graphical representation combines well-proven modelling constructs with new abstract representations for the concepts of a thinkLet and thinXel to improve the understanding of a collaboration process design [18]. Rules for the composition of these primitive modelling constructs were defined by adapting given workflow patterns [1]. An example of the graphical representation is shown in Figure 1, which shows a reusable process template called *Understand Process*. The goal of this process template is to provide the participant information about a process, e.g. the specific task of the group process. The activity to read this information is represented by the thinXel *Read Information*, which receive the data by the data parameter *Process Information*. A thinXel describes a binary logical design element that represents an atomic reusable activity of a participant that can be executed or aborted by the participant. In this example, the participant can decide if he or she needs more information in order to to understand the process. In this case, the thinXel will be aborted and the sender *Need Information* will be activated. The resulting signal of the sender will be forwarded by the element *Start Process* to initiate another process template, e.g. a process to discuss possible questions about the process.

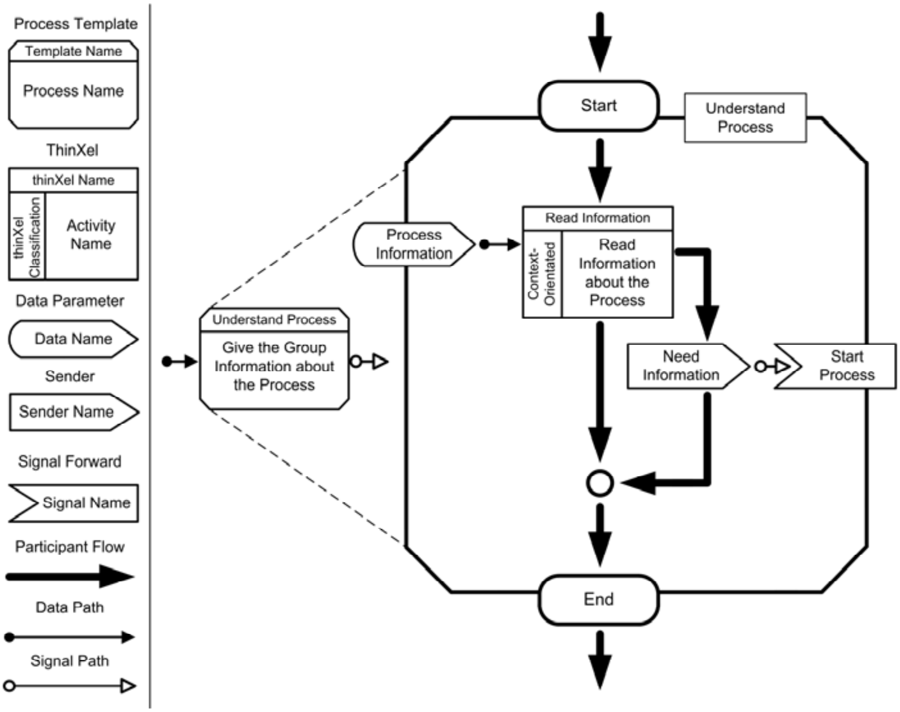


Fig. 1. Graphical representation of the collaboration modeling language

The same process template is shown in Figure 2 in a semantic notation using Extensible Markup Language (XML). Here, the thinXel *Read Information* is described by the XML tag *activity* and contains besides a short description a XML tag *parameter*, which defines a connection between the thinXel and the data parameter *Process Information*. The definition of this data parameter is defined in a superior tag. The semantical representation of a thinXel further includes a configuration block, which comprises formal information for the configuration combined with positive and negative examples for possible facilitation instructions. We will describe in Section 3.2 how we use this configuration block to provide knowledge for inexperienced users in order to adapt a collaboration process.

Our first application of the collaboration modelling language was a prototype of a generic groupware technology that links a group via the Internet and implements the activities of a collaboration process via a website [19]. We use the information about the configuration of a thinXel to provide a predefined web page that allows the participant to execute or abort the intended activity. Currently, the prototype provides no support for the configuration of the logical process model by inexperienced user. However, we believe that knowledge about facilitation can be documented and provided by the semantic process notation. For this reason, we analyzed the applicability of different approaches for instructional design to capture and transfer tacit knowledge of collaboration experts.

```
<processplan id="understandprocess" name="Understand Process">
   <description> In this phase the participants will read the process information
</description>
   <parameter id="processinformation" name="Process Information" type="string">
      <description>...defines the process information for the first phase
   </description>
      <config>
         <config:instruction> Please define in short sentences the task of the
process </config:instruction>
         <config:value/>
         <config:rule>
            <positive-example>
               In the next process we want to solve the task ...
               This task results from the situation ...
               It is necessary to solve this task because ...
            </positive-Example>
            <negative-example> The task is ... </negative-Example>
         </config:rule>
      </config>
   </parameter>
...
   <activity id="readprocessinformation" name="Read Information" type="read">
      <description> ... in this step the participants will read the process
information.</description>
      <parameter link = ‘processinformation’/>
      <config>
         <config:instruction> Please define a short instruction for the
participants: </config:instruction>
         <config:value/>
         <config:rule>
            <positive-example>
            Please read the process information.
            Click the button ‘OK’ to continue.
            To get more information about the process click the button ‘More
Information’</positive-example>
            <negative-example>Read Information</negative-example>
         </config:rule>
         <config:do-button-label value=’OK’/>
         <config:cancel-button-label value=’More Information’/>
      </config>
   </activity>
...
</processplan>
```

Fig. 2. Semantic representation of the collaboration modeling language

3 How to Transfer Facilitation Knowledge Using a Semantic Process Notation

In this section we will introduce our approach to transfer knowledge and skills for the configuration and execution of a collaborative process. According to Clawson et al. [6], different dimensions can be defined to describe the role of the facilitator in a computer-supported meeting. Our research concentrates on the transfer of knowledge that is needed to appropriately select a process that fits to a collaboration task and the intended outcome, to provide instructions that encourage the group to participate, to define clear and explicit instructions and information for the group and to explain the group how to use an selected technology to engender the intended process outcome.

3.1 Transferring Facilitation Knowledge for the Selection of a Collaboration Process

Similar to Briggs et al. [4], we agree that knowledge about collaboration can be packed and transferred by using a pattern approach like the design pattern thinkLet. However, the concept of a thinkLet only provides formal guidelines for the execution of a collaboration process documented in a paper-based handbook [23]. According to our modeling language for collaboration, we think that this knowledge should also be described by the semantical representation. Thereby we distinguish between a logical and a physical design of a collaboration process.

We define a logical process design as a process template that can be used to engender an abstract goal for a defined set of tasks and group characteristics. Including activities will be represented in an abstract description. A collaboration engineer can use a logical design to define processes independent from a specific task, which can be shared between different organizations via a process database. A physical process design is defined as a collaboration process for a specific task and group characteristics. In this process design, detailed descriptions of information about the task and needed instruction for the collaboration activities are defined. Therefore, an organization configures a logical process design for the given task by adapting the parameter and instructions of the process template.

Resulting from this distinction between process designs, the semantic representation needs to provide knowledge for the selection and adaptation of a collaboration process template for a certain context, group constellation and task. We adopted the pattern approach to provide information for the selection of collaboration process. Therefore, the semantical notation of a process template uses XML tags to capture and provide information. e.g. the name of a collaborative process by the tag *name*. To describe the situation a process design is intended to use, we used a tag called *context*, that contains the elements: *purpose* which captures the core of the process; *input* to describe possible needed input, e.g. a task or a list of contributions; *output* to describe the intended outcome of a process, e.g. ideas who solve a given task or a list of items sorted by categories. The tag *group* defines the group characteristic by the elements: *size* for the intended group size and *roles* for the needed skills of the group participants.

3.2 Transferring Facilitation Knowledge for the Configuration of a Collaboration Process

To support the adaptation of a logical process template to a physical design, parameter and facilitation instructions need to be configured for given task and group characteristics. The facilitator will need to define clear and explicit instructions and information for the group. The purpose of this instructional writing is to give instructions to the participants, which lead to the activities that have a well-defined function in the context of the group goal. These instructions should only contain those pieces of information that must be conveyed to the participants to perform the intended activities. We found guidelines for instructional writing in the research field of Controlled Natural Languages. Controlled Languages are used for the production of technical documentation or user-documentation [2], which uses a defined vocabulary and grammatical constructions that are also found in a natural language such as English. By analyzing the controlled language Simplified English [3] in regard to instructional writing, we identified twelve rules for instructional writing.

01. Give the necessary background information before the instruction.
02. Use a numbered list, if you want to present background information in a specific order. Otherwise use a bulleted list when the order is not important.
03. Use a sequence of activities, if you want to describe a process in the background information. A sequence of activities (A ' B ' C) is common when all participants perceive that activity C follows activity B and that activity B follows activity A.
04. Use words that any participant of a group understands.
05. Write in a friendly manner.
06. Address the participants of a group by using "you" or "your".
07. Describe only one atomic activity per instruction. Atomic activities are for example: read, write, select, think, draw or put.
08. Make an instruction as specific as possible.
09. Keep an instruction as short as possible.
10. Specify what the participant has to do when the intended task of an instruction is completed. If a reader asks, "Now what?", the instruction is not complete.
11. Use the active voice to define an instruction.
12. Write an instruction as a request using the polite word "please" before the verb.

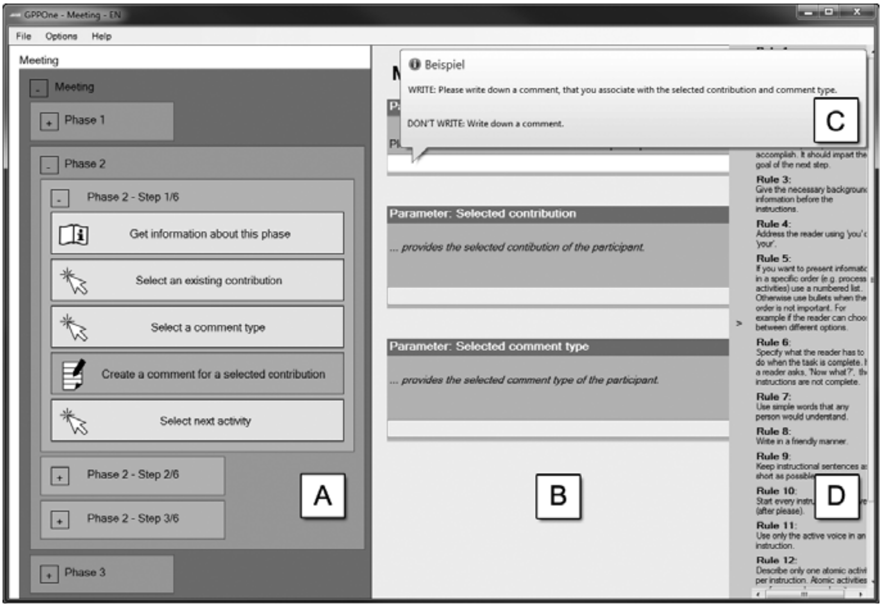
These rules were documented as guidelines for instruction-writing using the following structure:

> **Rule 7**: Describe only one atomic activity per instruction. Atomic activities are for example: read, write, select, think, draw or put.
>
> **Description**: If you put more than one activity into an instruction, the instruction will overwhelm the participant. In this case the participant can misinterpret an instruction. Therefore, the practitioner must present one activity per instruction at a time. That will lead the participant to complete one activity at a time.

WRITE:
1. Please select an issue from the cluster.
2. Please write down an idea that you associate with the selected issue.

DON'T WRITE:
Please select an issue from the cluster and then write down an idea that you associate with the selected issue.

During the adaptation of a logical process template to a physical design, an inexperienced user can use these guidelines to configure process parameters and facilitation instructions of the semantical notation. To support the user, the semantic notation of a parameter and activity captures and provides information in a XML schema for its configuration. We use the XML tag *description* to provide information about the parameter and activity in relation to the process design. The XML tag *configure* is used to provide information for the configuration of a parameter or activity (see Figure 2). The XML tag *config:instruction* is used by the process designer to instruct the user how to configure the parameter or activity. Further, the designer can provide rules (XML tag *config:rule*) with positive and negative examples (XML tag *positive-example* and XML tag *negative-example*) to support the inexperienced user.



WRITE: Please write down a comment, that you associate with the selected contribution and comment type.

DON'T WRITE: Write down a comment.

... provides the selected contibution of the participant.

... provides the selected comment type of the participant.

accomplish. It should impart the goal of the next step.

Rule 3: Give the necessary backgroun information before the instructions.

Rule 4: Address the reader using 'you' 'your'.

Rule 5: If you want to present informati in a specific order (e.g. process activities) use a numbered list. Otherwise use bullets when the order is not important. For example if the reader can choo between different options.

Rule 6: Specify what the reader has to do when the task is complete. I a reader asks, 'Now what?', th instructions are not complete.

Rule 7: Use simple words that any person would understand.

Rule 8: Write in a friendly manner.

Rule 9: Keep instructional sentences a short as possible

Rule 10: Start every instr (after please).

Rule 11: Use only the active voice in an instruction.

Rule 12: Describe only one atomic activi per instruction. Atomic activities

Fig. 3. Interface of a software prototype for the configuration of a collaboration process template

We used the resulting concepts to develop a software prototype (shown in Figure 3) that supports the user during the adaptation of a collaboration process template for a certain context, group constellation and task. This prototype uses the Extensible Markup Language to import a logical process design. By using the XML tags, the application provides a navigation tool (Figure 3 / Area A), which allows the user to change the focus between the different phases of a collaboration process. Parameter and instruction of a selected process phase can be adapted in the configuration tool (Figure 3 / Area B). This tool provides for each parameter and instruction a short description about the element and its relation to the collaboration process. By using the textboxes, the user can adapt the parameter and instructions of a process design. During this configuration, the prototype supports the user by two functionalities. A tooltip can be used to provide predefined positive and negative examples for the configuration (Figure 3 / Area C). Furthermore, the user can activate an information window that shows a summarized list of the defined guidelines for instructional writing (Figure 3 / Area D). An adapted process design can be exported into a semantic notation that can be used by a groupware technology to execute the collaboration process.

In the next section we will evaluate the presented pattern approach for facilitation knowledge transfer with expert interviews and a case study with inexperienced facilitators.

4 Evaluation of the Facilitation Knowledge Transfer Approach

We designed two experiments to test our facilitation knowledge transfer approach. Expert interviews were used to verify the defined rules for instructional writing. Furthermore a quasi-experiment was designed to test our assumption, that the given information of the logical process design can be used to support an inexperienced user during the adaptation of a logical to a physical process design.

4.1 Expert Interviews for Instructional Writing

With the expert interviews, we wanted to verify the defined rules for instructional writing by experienced facilitators. The goal of this part of the experiment was to gain personal experience of each individual expert to merge their knowledge and therefore confirm our approach described in Section 3.2.

We interviewed seven experts, four of them were male and three were female. The interview technique was chosen, because the experts were supposed to freely share their knowledge and opinion about the rules without constraints. Their experience with planning and executing group processes amounts between four to ten years. The amount of accomplished group processes between 20 and 200 processes, with a group size from three up to 150 participants. The group processes task differed between small meetings, innovation processes in companies, coaching of specific techniques, briefings with customers, conferences, and brainstorming sessions. The experts had expertise with homogeneous and mixed groups including students, employees, managers, engineers, and designers.

In a first phase, we asked general questions about their personal experience in designing a collaboration process and how to formulate process information and instructions. Almost all experts gave similar answers and reasons for the main steps of their process design, e.g. "*Always explain the goal before you start with process instructions*". They only differed in questions of detail. All of them used a structured and predetermined top-down approach for a collaboration process design. One expert offered a completely different process design with more degrees of freedom for the participants and no agenda. However, both possibilities were supposed to be successful regarding the goal of the group process. We used the knowledge of the experienced facilitators in order to design our experiment.

In a second phase, we presented our rules for instructional and descriptive writing and asked for feedback. Every expert had one or two rules which he or she described as not necessary for instructional design, for example "*Address the reader using 'you' or 'your'*". These differences occur because every expert has his or her own style during a collaborative process. We tried to get out the matching rules of the majority of all experts. In conclusion the most experts agreed that our rules are useful and necessary for the instructional design, which can be used during the adaptation of a logical to a physical process design. The list of rules matches the opinion of our experts.

4.2 Quasi-experiment with Inexperienced Users to Verify the Knowledge Transfer Approach

In a quasi-experiment, we tested our assumption that the approach transfers tacit knowledge that supports inexperienced users during the adaptation of a logical to a physical process design from the point of view of an inexperienced user. We designed an experiment that implements a configuration process of a process template using our software application. For this experiment, a professional facilitator developed three logical process designs using the graphical and semantical notation of our modeling language for collaboration. Each of the resulting process templates combine thinkLets of different collaboration patterns, e.g. thinkLets for the generation of concepts, the organization of concepts, and consensus building. Two of the logical process designs define a process for ideation and one is intended to structure a meeting process. Each process design provides information and example how to adapt a logical design for a specific task.

Scenarios. For the configuration of the logical process designs we used three scenarios. In the first scenario a process design for ideation should be adapted for the task *to generate event ideas for a university*. The second scenario uses also a process design for ideation but uses the task *to generate software ideas for a mobile device*. The last scenario uses the logical design for a meeting process of four students with the task *to prepare an event for a university*.

Participants. Thirty-six students from a university participated individually in this case study, 15 women and 21 men with different cultural background (Europe, Asia, North America and Africa). The participants' age ranged from 20 to 34 years. They had different experience with collaborative processes. Thirty-one had taken part in a group process before. Twenty of them had further experience with ideation techniques. Eleven participants had experience in facilitating a group process.

Experimental Design. The experiment was split into two phases. Upon arrival, the participants were trained in using the set of rules for instructional writing. The facilitator provided the objective of each rule. We demonstrate how to design an instruction or information using these rules and let the participants generate some instruction on their own. A facilitator gave feedback to these instructions.

In a second phase, the participants used the software application for the adaptation of the logical process designs in different specific scenarios. The participants received an introduction to the functionality of the software application. Each participant was given one out of three logical process designs. They were requested to configure the logical design in regard to one of the given scenarios. During this adaptation, the software application only provided the rules for instructional writing and the general information and example defined by the logical design. No verbal communication was allowed between the participants during this phase.

Questionnaires were used to collect the expertise of the participants with group processes as well as their experience with the configuration process using the software application. Here, closed questions were important in order to get quantitative analysis to support our hypothesis. Furthermore we compared the resulting physical designs with a process design that was adapted by an experienced facilitator in order to get a qualitative analysis.

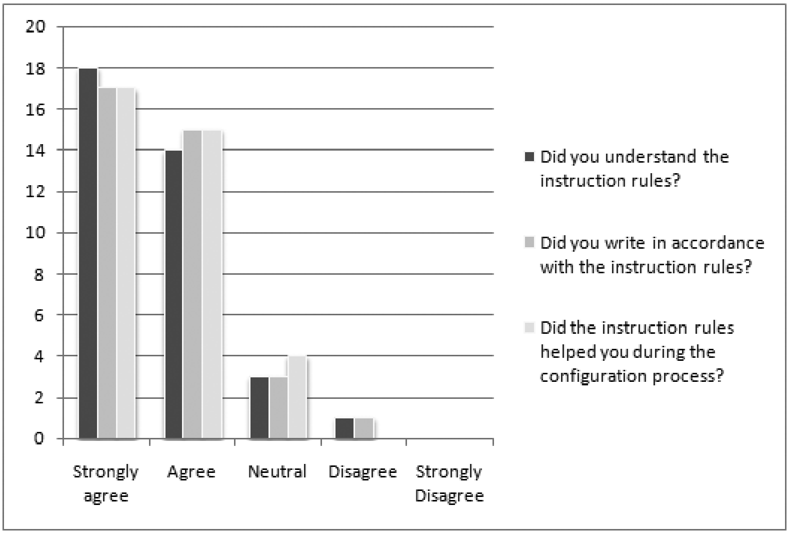


Fig. 4. Results of the evaluation of the instructional rules

Results. An analysis of the questionnaires in relation to the concept of instructional rules (see Figure 4) shows that most participants understood the rules for instruction writing. However, it seems that it was not easy for everybody to follow these rules during the configuration process. Some participants did not know what they had to do all the time. As a result, not everyone was able to create a functional group process. Most of the participants preferred the given structure of the logical design. They indicate that the rules for instructional writing and the examples provided helped them during the configuration process. Participants who have problems during this adaptation, claimed that the generic presentation of process parameters was not easy to understand in a certain context. In conclusion we can say that the instructional rules can be used to support participants during the configuration, but that the way we presented the parameters was not as successful as expected. An easier way to present the process information still has to be found.

In relation to the process understanding of the participants, the analysis shows that most participants understood the goal of the collaborative process. The software application helped them to get a good overview about the process. Most participants have no problems with the adaptation of the collaboration process. They created functional processes which they would also use in a real life scenario (see Figure 5). Most participants preferred the step by step guidance, but 16 participants would also like to have more degrees of freedom during the configuration process. Two participants disliked the step by step guidance. One of them was more advanced with the techniques used and group processes in general. Because of that he would have been able to create a group process with less guidance and more of his own creativity. The other participant misunderstood the process and the involved activities. That might be the reason why he did not like the guidance. In conclusion we can say that most participants were able to understand the group process and the technique. But there are still a few participants who were not able to create a decent group process despite of all guidance. It is necessary that the process design is able to respond to people who with more advanced abilities.

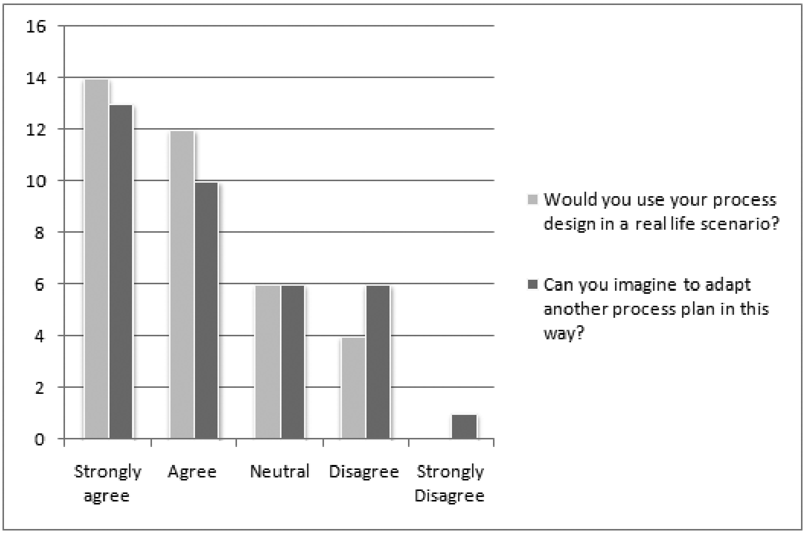


Fig. 5. Results of the evaluation of the process comprehension

Furthermore, we compared the resulting physical designs with a process design that was adapted by an experienced facilitator. The analysis shows a high similarity for most part of the adapted physical designs. However we found also differences in the process designs. In contrast to participants who invested a large effort in following the instructions, some participants just copied the positive examples. As a result, we were not able to create a situation where all or most participants produce a group process of the same level of quality. However, there seems to be a relation between the instruction design and the cultural background, the general background knowledge and the motivation of a participant. Participants with medium or higher experience with collaborative processes and the used techniques were able to create better group processes and needed less time to accomplish the task. These participants liked the concept of negative and positive examples. In context of the rules for instructional writing, the resulting process instructions fulfill some rules more often than others. It might be helpful to underline those rules in a training to make clear how important they are.

5 Discussion and Conclusions

This paper introduced a knowledge transfer approach for the adaptation of a logical process design by an inexperienced user. The approach transfers facilitation knowledge for the selection and configuration of a collaboration process and provides rules for instructional writing to support an inexperienced user in defining clear and explicit instructions. The result of the expert interviews described in Section 4.1 supports our assumption that the rules for instructional writing transfer tacit knowledge of experienced facilitators for the adaptation of a collaboration process from a facilitation expert's point of view.

A software application was used to evaluate the knowledge transfer approach in a quasi-experiment with inexperienced participants. This experiment verified that the knowledge transfer approach is applicable to transfer tacit knowledge of experienced facilitators to inexperienced users from an inexperienced user's point of view. Furthermore it verified the use of the transfer approach for the adaptation of a logical to a physical process design. The results suggest support for our assumptions, but we still found differences in the quality of the resulting process designs. So we can say that it provides necessary information to support inexperienced users, but that this information is not sufficient.

A number of limitations exist in this experiment. We used three different group processes to provide a basis for the experiment (two processes focus on ideation; one defines a meeting process), which limited the results to those group processes. Limitation is also given by the use of a small number of students in a laboratory environment. A larger sample size should reduce the effect of possible outliers in the measured data on the result. Furthermore the lab experiment should be tested in a real-world environment in order to approve the results in this work.

The introduced knowledge transfer approach provides interesting concepts that can be used by a machine-readable process description for collaboration. However, research is needed to extend and verify our approach. We considered a large number of instruction rules, but our approach cannot be seen as a complete set of instruction rules. Furthermore, a better process overview should be considered to overcome the difficulty that some people did not understand the group process. The step-by-step guidance is preferred by inexperienced users. But as soon as people become more experienced they want more degrees of freedom to bring in their own creativity and style. Then they do not want to follow a strict group process construct. A balance has to be found to bridge the needs from very inexperienced facilitators to more experienced facilitators. The first step of bridging the needs could be to enable the user to fade the rules in and out. In the future, perhaps less information should be shown the more experienced the user is, for example, a three-step instruction could be merged to a single-step instruction.

References

1. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distributed and Parallel Databases 14, 5–51 (2003)
2. Adriaens, G., Schreors, D.: From COGRAM to ALCOGRAM: Toward a controlled english grammar checker. In: Proceedings of the 14th Conference on Computational Linguistics (1992)
3. Aecma simplified english. Technical Report PSC-85-16598, The European Association of Aerospace industries, Bruxelles (1995)
4. Briggs, R.O., de Vreede, G.-J., Nunamaker Jr., J.F.: Collaboration engineering with thinklets to pursue sustained success with group support systems. Journal of Management Information Systems 19(4), 31–64 (2003)
5. Briggs, R.O., Kolfschoten, G.L., de Vreede, G.-J., Dean, D.L.: Defining key concepts for collaboration engineering. In: Proceedings of the Americas Conference on Information Systems (AMCIS), pp. 121–128 (2006)
6. Clawson, V.K., Bostrom, R.P., Anson, R.: The role of the facilitator in computer-supported meetings. Small Group Research 24(4), 547–565 (1993)
7. Csikszentmihalyi, M.: Creativity: Flow and the Psychology of Discovery and Invention. Harper Collins Publishers, New York (1997)
8. Dennis, A.R., George, J.F., Jessup, L.M., Nunamaker Jr., J.F., Vogel, D.R.: Information technology to support electronic meetings. MIS Quarterly 12(4), 591–624 (1988)
9. Dennis, A.R., Wixom, B.H., Vandenberg, R.J.: Understanding fit and appropriation effects in group support systems via meta-analysis. Management Information Systems 25(2), 167–193 (2001)
10. DeSanctis, G., Gallupe, R.B.: A foundation for the study of group decision support systems. Management Science 33(5), 589–609 (1987)
11. DeSanctis, G., Poole, M.S.: Capturing the complexity in advanced technology use: adaptive structuration theory. Organization Science 5(2), 121–147 (1994)
12. Diehl, M., Stroebe, W.: Productivity loss in idea-generating groups: tracking down the blocking effect. Journal of Personality and Social Psychology 61(3), 392–403 (1991)
13. Edmondson, A.: Psychological safety and learning behavior in work teams. Administrative Science Quarterly 44(2), 350–383 (1999)
14. Hollingshead, A.B.: The Workflow Reference Model, Workflow Management Coalition (1995)
15. Janis, I.: Groupthink. Houghton-Mifflin, Boston (1982)
16. Karau, S.J., Williams, K.D.: Social loafing: A meta-analytic review and theoretical integration. Journal of Personality and Social Psychology 65(4), 681–706 (1993)
17. Knoll, S.W., Chelvier, R., Horton, G.: Formalised online creativity using thinXels. In: Proceedings of the 10th European Conference on Creativity and Innovation (2007)

18. Knoll, S.W., Hoerning, M., Horton, G.: A design approach for a universal group support system using thinkLets and thinXels. In: Proceedings of the Group Decision and Negotiation Meeting (2008)
19. Knoll, S.W., Hoerning, M., Horton, G.: Applying a thinkLet- and thinXel-based group process modeling language: A prototype of a universal group support system. In: Proceedings of the 42nd Hawaii International Conference on System Sciences (2009)
20. Kolfschoten, G.L., den Hengst-Bruggeling, M., de Vreede, G.-J.: Issues in the design of facilitated collaboration processes. Group Decision and Negotiation 16(4), 347–361 (2007)
21. Kolfschoten, G.L., Santanen, E.L.: Reconceptualizing generate thinklets: The role of the modifier. In: Proceedings of the 40th Hawaii International Conference on System Sciences (2007)
22. Kolfschoten, G.L., de Vreede, G.-J., Briggs, R.O.: Computer aided pattern-based collaboration process design: A computer aided collaboration engineering tool. In: Haake, J.M., Ochoa, S.F., Cechich, A. (eds.) CRIWG 2007. LNCS, vol. 4715, pp. 165–172. Springer, Heidelberg (2007)
23. Kolfschoten, G.L., de Vreede, G.-J.: A design approach for collaboration processes: A multimethod design science study in collaboration engineering. Journal of Management Information Systems 26(1), 225–256 (2009)
24. Kolfschoten, G.L., Lee, C.: Self-guiding group support systems: Can groups use GSS without support? In: Proceedings of the 43rd Hawaii International Conference on System Sciences (2010)
25. Macaulay, L.A., Alabdulkarim, A.: Facilitation of e-meetings: State-of-the art review. In: Proceedings of the IEEE International Conference on E-Technology, E-Commerce and E-Service, pp. 728–735 (2005)
26. Mittlemann, D.D., Briggs, R.O., Murphy, J., Davis, A.: Collaboration technology, computer-based collaboration-support products. In: Briggs, R.O., Antunes, P., de Vreede, G.-J., Read, A.S. (eds.) Groupware: Design, Implementation, and Use, pp. 305–317. Springer, Heidelberg (2008)
27. Niederman, F., Beise, C.M., Beranek, P.M.: Facilitation issues in distributed group support systems. In: Proceedings of the 1993 Conference on Computer Personnel Research, SIGCPR 1993, pp. 299–312 (1993)
28. Nunamaker Jr., J.F., Dennis, A.R., Valacich, J.S., Vogel, D., George, J.F.: Electronic meeting systems to support group work. Communications of the ACM 34(7), 40–61 (1991)
29. Nunamaker Jr., J.F., Reinig, B.A., Briggs, R.O.: Principles for effective virtual teamwork. Communications of the ACM 52(4), 113–117 (2009)
30. Pedro, A., Ho, P.,, T.: Carrico, Luis.: A gdss agenda builder for inexperienced facilitators. In: Proceedings of the 10th EuroGDSS Workshop (1999)
31. Poole, M.S., Hollingshead, A.B.: Theories of Small Groups. Sage Publications, Thousand Oaks (2005)
32. Schumann, S.: The IAF Handbook of Group Facilitation: Best Practices from the Leading Organization in Facilitation, Jossey-Bass, San Francisco (2005)
33. Shannon, C.E.A.: Mathematical Theory of Communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
34. Terveen, L.G.: Overview of human-computer collaboration. Knowledge-Based Systems 8(2-3), 67–81 (1995)
35. de Vreede, G.-J., Briggs, R.O.: Collaboration engineering: Designing repeatable processes for high-value collaborative tasks. In: Proceedings of the 38th Hawaii International Conference on System Sciences (2005)
36. Wong, Z., Aiken, M.: Automated facilitation of electronic meetings. Information & Management 41(2), 125–134 (2003)