Abstract

Research has demonstrated the added value of Group Support Systems (GSS) when they are applied under the right circumstances. However, the design of a GSS session remains a complex task and is vital for successful intervention using the technology. GSS supported sessions are dynamic and mostly unpredictable events, and preparation is therefore difficult. This paper presents a suite to support novice facilitators in the design of GSS sessions. The suite contains several tools and supports not only the design process of a GSS session but also the learning process of a novice facilitator during the preparation. The paper will explain the theoretical basis of the suite, the tools within the suite and a first evaluation of the added value of this suite for novice facilitators.

1. Introduction

Research has shown that under the right circumstances, Group Support Systems (GSS) can increase the productivity of meetings, compared to manual meetings. [14, 15, 27] Groups using GSS often use support of a facilitator. The design or preparation of sessions supported by GSS is a complex task. [10] Not only the group dynamics, and unpredictability of a group of people collaborating, pose complexity to this task. The complexity of preparation is also increased because the GSS can be used in many ways and with very different results. With preparation or design we do not mean just an agenda. Preparation involves a strategy to design a collaboration process; it includes the agenda, questions asked, task and process descriptions, and estimated results.

The importance of preparation of collaboration processes and especially of GSS supported sessions is indicated in literature in several ways. Nunamaker et al, [23] describe preparation as a critical success factor, while Vreede et al, [26] describe bad preparation as one of the important reasons for failure of GSS meetings. Both participants and facilitators consider preparation or design a very important task of a facilitator. [1, 8, 10,25]

The aim of this paper is to design and test a prototype of a suite that supports the design of a collaboration process. The suite will address the complexity of GSS session design enabling easier design of a GSS session, and support of the learning process of novice facilitators. Since it is a learning suite, it should eventually enable the facilitator to design without the suite. Testing the prototype will help us find challenges, complex aspects, and the needs, effective support tools and concepts, for support of facilitators, both in their learning process and in their design efforts.

The suite is a prototype and is built using a set of 10 basic thinkLets to reduce the complexity. The ten are chosen to provide enough support for a general meeting. The suite can be used for any set of thinkLets or facilitation building blocks, but a basic set is used to test the concept. The choice of these 10 thinkLets is influenced by the research on thinkLet modules and frequency of thinkLet use, [19] but also on expert opinion. The choice for the design of a software suite instead of a manual or guidelines has the following reasons:

- Structure: A computer program is more structured. Because of the layers in a program it is possible to make more overview of information.
- Selective information: More than text a computer program enables the user to skip information he already knows and repeat or rehearse other information
- Richness of media: in a computer program pictures, models, text, sound, video etc, can be combined, use of colour and schemes is more logical, easier and cheaper.
- Interaction: A computer program is more interactive and enables an active learning approach. [17]

In the remainder of this paper we will first describe the previous efforts in design support. Next the learning approach, design process and the previous research will be
explained as the background of the suite. The fourth section will describe the suite. In section five the evaluation of the suite and its results are described. After presenting the limitations, the paper will describe the contribution of this research and indicate directions for use and further research.

2. Tool support for GSS session design

To enable organizations to profit form the benefits of GSS, design support is required [13]. Dennis et al describe that two aspects are important in design; a good match of the task with the components of the GSS and appropriate use of those components. Unfortunately, the both are mostly available as expertise of facilitators, and not documented. The suite will attempt to combine information about the GSS components and about methods to use these components.

Since design of GSS sessions is the main obstacle for success of GSS sessions, there already exist some support tools and methods for the design phase. First of all some of the group support systems offer an agenda builder. The agenda builder enables the facilitator to combine the components of the GSS and to fill in the time schedule. However, the agenda builder does not support the choice of the components, or the different ways in which they can be used. Neither does it indicate how much time is required per step. Antunes et al [1,16] created a system that supports the choice of GSS components. The system enables the choice of a GSS component based on several aspects including process, task and strategy. The result is an activity coupled to GSS components, for example “expose and clarify selection criteria with GroupSystems™ Group Outliner”.

However, the components of a GSS can be used in many ways, with many different configurations. The system of Antunes et al [1,14] does not indicate a method to use these GSS components, or the specific configuration to use, or what time to spend on different steps. This is relevant because for instance the Group Outliner component of GroupSystems™ (Workgroup Edition 2.1) has 19 configurable features, for a total of 524,288 possible combinations. Different configurations have a large impact on the group process. [5] Therefore, supporting the choice for a component and activity alone is not sufficient. We need to specify how the GSS should be used for a specific situation.

Briggs et al, [5] describe an approach using thinkLets. ThinkLets are documented techniques to create repeatable, predictable patterns of collaboration. They describe the GSS component (tool), its configuration and a script for the facilitator on how to use the component and how to instruct the group. The first experience with thinkLet use through novices shows that with a short training, practitioners can recreate a pattern in the collaboration of a group [6]. Recent research has indicated that the choice of a thinkLet (instead of a GSS component) poses a new problem. This choice does not only depend on criteria such as the available time and the task, it also depends on the previous thinkLet and the next thinkLet. [19] There are both configuration and process issues that make some combinations impossible or difficult and others very successful. The suite presented in this paper will not support the choice of a GSS component, but the choice of a thinkLet.

The support of the choice for a thinkLet can be based on the main characteristics of a thinkLet; insight in implications of the combinations of thinkLets is drawn from research on patterns in thinkLet use [19]. Time indications to support planning were not yet available. Therefore experts using a Delphi method [21] made estimates of the time required to perform a thinkLet. To support the design of a collaboration process, a design process should be identified. In the next section the design process will be described. A last aspect of the design of a collaboration process is documentation of that design. A format for design documentation based on [4] is used for this purpose.

3. Background

The suite will be based on earlier thinkLet research to support the design of a predictable process resulting in a description of the exact use of a component of GroupSystems™. In addition, the suite will be based on various learning principles and learning styles. The next sections will describe the learning principles and the thinkLet design approach

3.1 Learning principles

The suite designed in this research aims to enhance and speed up the learning process of novice facilitators in designing GSS sessions. We have created a learning environment that should accommodate a variety of learners. The overall learning approach of the suite assumes that learning is both an active process and that learning is enhanced by experience. Although the concept of experience and analysis as important steps in the learning cycle is already recognized by Kolb, [18], in new multimedia environment it is extended in the theory of constructivism. Constructivism is a philosophy of learning founded on the premise that, by reflecting on our experiences, we construct our own understanding of the world we live in [7]. Constructivists view learning as the result of mental construction. Students learn by fitting new information together with what they already know. People learn best when they actively construct their own
understanding. Learners are encouraged to invent their own solutions and to try out ideas and hypotheses. They are given the opportunity to build on prior knowledge. Based on this constructivist view the researchers have developed a suite in which learners can experiment and try out designs for GSS sessions using thinkLets, find out how to sequence thinkLets appropriately, and get immediate feedback on their GSS designs. In order to accommodate learners to memorize thinkLets a memory tool has been developed providing learners with simple drill and practice exercises.

The suite consists of 3 interrelated but independent learning processes that learners can use in any order. The suite accommodates the learner in choosing his learning path, rather than the software prescribing it. Some might start memorizing thinkLets, others might prefer to begin designing the GSS session and see what happens. The learner should be the centre of the learning process, actively constructing his understanding.

The first learning process aims at creating a cognitive library of thinkLets and basic knowledge on GSS and facilitation. It helps facilitators to get familiar with and memorize thinkLets and facilitating skills. However, memorizing the thinkLets is not a compulsory first step for a learner, during the design of a GSS session, information can be looked up. Short videos clips and summaries of thinkLets and their mnemonics will support the learner in memorizing the thinkLets. Mnemonics aspects can be found in the names of thinkLets, in the pictures that accomplish them, in abbreviations, etc. Mnemonics are based on the functioning of the human brain, as described by Bjork and Bjork. [3] In time, the learner will create a cognitive library of thinkLets. This will support him, as during a GSS session a facilitator might have to switch from one to another thinkLet unexpectedly according to changing needs of the group. The second learning process supported by the suite aims at the understanding of the thinkLet concept and relations between the components of the concept. To give the novice insight in these aspects, the researchers created a set of example models accompanied with instruction videos. The models describe the relations between the different components of the thinkLet concept and give insight in real-life situations.

In the third learning process supported by the suite, the novices can experiment with the design of consecutive thinkLets, providing appropriate feedback at each of the phases of the design and the choices made by the novice facilitator. It is the most challenging part where learners can apply or develop their basic skills and find out the implications of different design approaches, enabled by the predictability of thinkLets.

3.2 ThinkLet Design

The design process (see Fig. 1) is part of a general problem solving process, as described for different purposes as for instance: creativity [11], problem solving [2, 22], decision making [24], and systems analysis and design [9], among others. The process starts with an analysis that will reveal the requirements to the problem. The thinkLets provide the possible solutions. These should be chosen and combined into a design, which is then validated in respect to the requirements. From this process especially the last two stages, the choice and the validation, are supported. The solution requirements (phase 2) can be analysed with a checklist. The list contains both task and group aspects of a collaboration process that are important during the design. The analytical interviewing skills required for this analysis are not considered. The thinkLets, collaboration building blocks, have an important role. They provide pre-defined solutions. This changes the focus of the facilitator from finding solutions to choosing the best (combination of) pre-defined solution(s) (thinkLets). The thinkLet design approach is based on five patterns of collaboration:

- Diverge: Move from having fewer to having more concepts.
- Converge: Move from having many concepts to a focus on and understanding of a few deemed worthy of further attention.
- Organize: Move from less to more understanding of the relationships among concepts
- Evaluate: Move from less to more understanding of the benefit of concepts toward attaining a goal relative to one or more criteria.
- Build Consensus: Move from less to more agreement among stakeholders so that they can arrive at mutually acceptable commitments [6,19,28].

These patterns describe what can be predicted about the thinkLet; if a divergence thinkLet is used we can estimate that the group will produce ideas or comments. We can also predict something about the structure of the data that are produced.

ThinkLets are originally documented in a thinkLet book. [5] The complete documentation of the thinkLets should be used as preparation for the facilitation of a session. Not all this information on thinkLets is required during the design phase to inform the choice of a thinkLet. The choice for a thinkLet can be based on a summary of the documentation. However, some additional information about each thinkLet is required as well. The summaries in the suite provide the following information on each thinkLet to provide the predictability, GSS component and use, and support for preparation and design:
The name serves as a mnemonic for the thinkLet. It is a name that intends to cover the description of the process. Existing techniques are hard to identify or remember by their name, think for instance of “categorized brainstorming”. Mnemonic aspects are therefore important for identification of the thinkLets.

As described before the pattern of collaboration is very important. It enhances the predictability of the thinkLets.

The structure of the result of a thinkLet can also be defined; it is valuable information for the predictability of the thinkLet.

The characteristics describe the extra value of a thinkLet. This can be a special feature but it can also be a limitation to its use, for instance a thinkLet is only suitable in small groups, or very efficient for large groups.

The GSS component, configuration and script are summarized. This will give the user an idea about the implementation of the thinkLet. This summary is sufficient to recreate the pattern of collaboration and the result; it will also answer to the characteristics. Details about these aspects can be found in Vreede et al. [28]

The time estimates for the thinkLet are based on experience. They vary for large or small group size and task size. The estimates are vital for novices to support them in creating a timeframe for the meeting.

The last aspect of the summary is the input that is required. The input can be a data set from a previous step, or for example a set of criteria. An example of a thinkLet summary for LeafHopper can be found in fig. 2 at the right side. The suite furthermore uses two additional building blocks related to the thinkLet concept: the module and the transition. [19,20] The transition is an interface between thinkLets. It describes how to create a smooth transfer from one thinkLet to the next. It also describes the advantages and problems with respect to certain combinations of thinkLets. The module is a combination of thinkLets and transitions in a sequence that creates, modifies or evaluates one data set. A data set is a set of concepts on a topic that can be modified and extended. However, as soon as a new data set is made from scratch on a new subject, a new module is build. [19,20]

If novices can better and faster design GSS sessions, it will be easier to use the complex but very effective GSS technology successfully. The previous sections explained the learning approach used, the design approach used with the thinkLet concept. The following requirements for the suite can be distilled from these descriptions:

1. The suite should be easy to use
2. It should give insight in the choice of a method of GSS use and the specific GSS component and configuration of the GSS in this case
3. It should have predictable results, a novice does not have experience to rely on, therefore the methods offered should be reliable to create the expected process and results
4. It should give insight in the implications of a design. Not only the single thinkLets should match the requirements of the task, also the combinations of thinkLets should make sense, as described above and in Kolf Schoten et al [19]
5. It should indicate the time required for a thinkLet, to enable planning of the agenda.
6. It should format the documentation of the design and support the design process to add relevant content to it.
7. It should enhance and speed up the learning process of novices.

4. The Suite

To reduce the complexity of GSS session design and enables fast training of novices the suite has four tools, which each support a specific part of the learning and design process. The first two learning phases, knowing and understanding the concept are supported by respectively the memory and flexibility tool. The design tool supports the design process, and enables feedback to the combinations of thinkLets and choices in the design process. The control tool enables validation. The tools will contain the following information and functionalities, each provided by a button
in the main screen of the suite. Some functionality exists of small programs; others are images, models, videos or text.

- The memory tool contains basic thinkLet content information and supports memorization and communication of the thinkLet summaries.
- The flexibility tool contains the models of the system and supports overview of the big picture. It gives insight in the possible combinations of thinkLets and therefore enables flexibility in altering the design.
- The design tool provides practical information and supports the largest part of the design process, especially the choice of the thinkLets based on several criteria sets. The choice tool and library are shown in fig. 2. In the choice tool the design is made. First a starting point is chosen. The starting point can for instance be “from scratch” or a list of ideas. The starting point narrows the set of possible thinkLets. Some thinkLets require specific input. Next a thinkLet can be chosen. By clicking on the thinkLet, the summary appears in the library, additional information on the thinkLets can be found in the instructions to the library. To narrow the choice, one of the criterion sets can be used. The criterion sets are based on the patterns of collaboration, the results of the thinkLets and their characteristics, each representing an aspect of predictability of the thinkLet. Next the user can indicate factors that enable the time estimation and choose the thinkLet that now appears in the design at the left side of the screen. The choice of a next thinkLet is narrowed first by the previous thinkLet. For each following thinkLet the color code indicates the value of the combination: yellow is best, green is ok, orange indicates risks or possible problems. The criteria can again be used to narrow the choice. The criteria represent the predictable aspects of a thinkLet; the pattern of collaboration, the result structure of the thinkLet and its characteristics.
- The control tool provides evaluation information about time frames, group dynamics, balance in typing and discussion and roles in the session. It enables the user to evaluate the total design.

5. Evaluation

The evaluation of the prototype has two purposes: First, the
suite should be evaluated with respect to its requirements. Most aspects of the requirements are simply included, therefore those aspects will be evaluated on quality and efficiency of the designs made by novices. The ease of use is the only requirement in section 1 that is not identified as a component of the suite. The second purpose of evaluation is addressing the learning process of novices and the success of their design efforts.

5.1 Design of the evaluation

The evaluation has an exploratory nature. 12 novice facilitators, who were divided in 3 experience categories based on the number of training hours they received and their experience in facilitation, used the suite:

- Level 1: 2-4 hours, no facilitation experience, some with minor technical facilitation experience.
- Level 2: 8-10 hours, some with technical facilitation experience, and no facilitation experience.
- Level 3: approximately 20 hours, limited facilitation experience.

A technical facilitator, chauffeur or technographer supports a facilitator operating the GSS. There were 2 evaluation sessions, one with 8 participants, one with 5 participants. The groups performed cases that described a detailed problem that could be solved in a collaboration process. The cases in both situations were different. Despite the effort to make the cases equally difficult, in the first group one case was significantly harder to design. One participant participated in both sessions.

Each group was introduced to the suite in less than half an hour. Furthermore, the novices were trained in GSS use, facilitation skills and in manual design, ranging from a short introduction to a full course, depending on the amount of training hours. All participants are undergraduate students from Delft University of Technology, ranging from 2nd to 7th education year. Each student had approximately 6 hours, to design 3 workshops. Not all students succeeded in designing 3 workshops within this time frame. The workshops were designed based on cases that described a goal, a group, and the contours of the task of a previous collaboration process facilitated by facilitators of the Delft University of Technology.

To evaluate ease of use we used the Technology Acceptance Model (TAM) model construct of Davis [12]. We measured the ease of use of the different tools and functionalities of the suite using the average of 6 questions measuring ease of use. Since only the design tool and control tool were meant to support the design process, and the subjects of the case study had no time to first learn the whole concept, we will present only those results. A 7-point scale is used here, as in the TAM model questionnaires; other constructs of TAM are not relevant since we are not yet interested in acceptance.

To evaluate the quality of design it was hard to follow a straightforward path. The following criteria were used and assessed on a 1 to 10 scale. This scale differs from the ease of use scale because the assessment of students in the Netherlands is always done on a 10-point scale and therefore the results were comparable with the results of previous student designs of the same cases. Two teachers of a facilitation class, both experienced facilitators, assessed the quality of the designs on the following criteria:

- The session is likely to reach the goal that is set
- The content of the questions per activity is unambiguous and clear
- There is balance in the session between typing and talking
- The session is adapted to the groups experience, competence, authority, etc.
- The session can be executed within the timeframe.
- The steps in the process have a logic order and contribute to the goal
- There will be some level of consensus at the end of the session.

With the quality construct we will measure a learning effect, indicated by an increase of quality over sessions designed after each other. Furthermore we will measure the efficiency of the tool support indicated by relative higher quality compared to a shorter training time. Another indicator for a learning effect is a decrease of the time spent to finish a design. To measure this all participants clocked the time they spent on their design.

The last comparison is made between the results of designs made for a specific case. This case was both used by a group of students that used the thinkLet book [5] and a group that used the support of the suite. The comparison will include the similarity and predictability of manual and suite supported design.

5.2 Ease of use.

The table (table 1) shows the ease of use of the tools and their functionalities. Like the design tool the control tool has four functionalities, but they are similar in use and therefore measured only once. Summarized we can state that the design and control tool are easy to use, especially the library function. The results were compared with indications of the most and least used tools. The control tool and checklist were hardly used during the experiment. The
choice tool and library were used most. Use of the design format was obliged to be able to compare results.

<table>
<thead>
<tr>
<th>Tool/Functionality</th>
<th>Average</th>
<th>Standard deviation</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design tool</td>
<td>5.44</td>
<td>0.78</td>
<td>11</td>
</tr>
<tr>
<td>Checklist functionality</td>
<td>5.98</td>
<td>0.63</td>
<td>10</td>
</tr>
<tr>
<td>Documentation func.</td>
<td>5.15</td>
<td>1.33</td>
<td>11</td>
</tr>
<tr>
<td>Library functionality</td>
<td>6.18</td>
<td>0.65</td>
<td>10</td>
</tr>
<tr>
<td>Choice functionality</td>
<td>5.36</td>
<td>0.69</td>
<td>11</td>
</tr>
<tr>
<td>Control tool</td>
<td>5.29</td>
<td>1.04</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 1: Ease of use, 7 point scale, 1 = very difficult to use, 7 is very easy to use.

### 5.3 Learning curve

The learning curve with use of the suite can be distilled from the quality of cases designed after each other. Table 2 indicates the quality of the first, second and sometimes third design made by novices with different training history. Interesting is that the novices get up to speed and outperform more experienced people. The latter however already adapted their own design method, and find it hard to use the new approach.

<table>
<thead>
<tr>
<th>Training time</th>
<th>20 h level 3</th>
<th>8-10 h level 2</th>
<th>2-4 h level 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Quality n</td>
<td>Quality n</td>
<td>Quality n</td>
</tr>
<tr>
<td>1</td>
<td>6.5</td>
<td>6.3</td>
<td>5.6</td>
</tr>
<tr>
<td>2</td>
<td>5.9</td>
<td>6.7</td>
<td>5.7</td>
</tr>
<tr>
<td>3</td>
<td>6.2</td>
<td>6.9</td>
<td>7.3</td>
</tr>
</tbody>
</table>

Table 2: Average quality of design of following designs assessed on a 10 point scale, distinguished by training time.

### 5.4 Efficiency

The efficiency is indicated by the time spent for a design. The time spent on design does decrease over time only for more experienced novices. The novices with low training time did not get faster in their design. However the general decrease in design time could also be caused by the fact that motivation and energy were lower at the end of the day. See table 3.

<table>
<thead>
<tr>
<th>Training time</th>
<th>20 h level 3</th>
<th>8-10 h level 2</th>
<th>2-4 h level 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Time n</td>
<td>Time n</td>
<td>Time n</td>
</tr>
<tr>
<td>1</td>
<td>79.17</td>
<td>93.50</td>
<td>108.00</td>
</tr>
<tr>
<td>2</td>
<td>83.33</td>
<td>86.20</td>
<td>70.00</td>
</tr>
<tr>
<td>3</td>
<td>37.50</td>
<td>60.00</td>
<td>105.00</td>
</tr>
</tbody>
</table>

Table 3: Average time spend on design of following designs measured in minutes, distinguished by training time.

<table>
<thead>
<tr>
<th>Group</th>
<th>Training time</th>
<th>Average quality</th>
<th>Standard deviation</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>level 3</td>
<td>≈ 20+ hours</td>
<td>6.23</td>
<td>0.55</td>
<td>16</td>
</tr>
<tr>
<td>level 2</td>
<td>≈ 8/10 hours</td>
<td>6.58</td>
<td>0.44</td>
<td>5</td>
</tr>
<tr>
<td>level 1</td>
<td>≈ 2/4 hours</td>
<td>6.03</td>
<td>0.86</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 4: Average quality of all suite designs versus training time in 3 distinguished groups. Quality is measured on a 10-point scale.

Efficiency is more visible in the training time itself. Table 4 shows that a group of experienced designers with 20 hours of training time score just 0.2 points higher on our quality scale than the novices with 2-4 hours training. The novices with slightly more training in design and some feeling for sessions scored 0.35 points higher. The higher score for the level 2 novices might be due to more specific training on design, and more participants with chauffeur experience.

### 5.5 Manual vs suite supported design

In this evaluation the same case is designed manual and suite supported by different novice facilitators. All manual facilitators are level 1 facilitators (indicated in table 5 as M1 to 10), for the suite facilitators the experience level is indicated (Table 5: S 1 to 7 L1= level 1, L3 is level 3). In table 5 we see the abbreviations of the thinkLets they chose for the sub tasks. In the abbreviations the first letter indicates the pattern of collaboration e.g. D= Diverge, the second and third letter abbreviate the thinkLet name. Thus DLH means Diverge, LeafHopper. The manual facilitators worked with the thinkLet book, and thus had more thinkLets to choose from. Although the end and start of each case is very similar, (the first DOP thinkLet in some of the sessions considers introduction) the convergence/organizing steps are different in the suite designs than in the manual designs. Both approaches are possible to solve the problem. The pattern that appears in the manual sessions contains a thinkLet that in the suite is described as difficult for novices and time consuming. This is probably the reason why the suite facilitators did not use it.
The results indicate patterns, for instance many start with DFB, followed by either CFF (7 x mostly in manual sessions, light shaded) or a combinations of OPS and OCO (6x mostly in suite supported sessions, darker shaded). The sessions often use DLH in the end, in both manual and suite supported sessions (dark shaded). There are two recurring patterns (underlined) in design made with the suite and in the manual designs. The one in the suite designs occurs relatively more (43%) than in the manual designs (30%). It seems that the suite gives a more predictable outcome than the manual approach. This is of course also due to the fact that fewer thinkLets are available for the suite users, but also through the information about combinations of thinkLets supplied by the suite. Differences in the patterns with suite support are more likely due to different interpretation of the decomposition of the process, than due to the choice of a thinkLet for each step within this decomposition.

6. Limitations

The evaluation does give promising results, but given the limitations below, more research is required to further confirm the results. However, since the system is a prototype the aim of the evaluation is to find aspects of the suite that can be improved, but more important it showed us which aspects of the design process and learning process require

this kind of support. The following limitations should be taken into account:

- Only the manual group worked for a grade, indicating a difference in motivation of the participants.
- The case description replaced the interview with the problem owner therefore the analysis support was not used
- The timeframe for the students to design the cases was short. More time for first contact and exploration of the suite, and feedback after each design is expected to increase the learning experience
- Not all participants filled in all aspects of the questionnaire because they did not use some of the tools.
- The time frame for both experiments was short compared to the exercise; participants therefore took little effort to explore the suite and instructions and only used the choice tool and library.
- Introduction was different for participants, some already had GSS and facilitation experience, others were complete novices.

• First group (mixed levels)
  ○ One case was significantly more difficult to design because it required steps that required the identification of different modules, a rather difficult concept for the novices.
  ○ In this experiment the case descriptions were quite suggesting in the required sub steps.

• Second group (mixed levels)
  ○ One participant has done both experiments.
  ○ The design format is changed; it explicitly asks for precise questions, one of the quality measures.
  ○ Cases were less detailed or suggesting in approach and sub steps.
  ○ Introduction for complete novices has been 4 hours and contained different information than in the first case.

7. Conclusions and further research

Based on the requirements in section 3 the suite provides a nice prototype for effective and efficient support of novice facilitators. As the questionnaire indicated the suite is relatively easy to use for design purpose, for training the ease of use is not addressed. It would be interesting to see the results of the suite in a more step-by-step learning process. The suite does support the design process and the learning process; it gives insight in the choice with transition information, and it provides advice and feedback for implementation. The predictability of the results is more explicit than in the thinkLet book, this has both advantages, there are less choices to make, and disadvantages, the suite
does not support deviation from its suggestions. The suite has newly developed time estimates and suggests the novice to document the design in a format. The evaluation shows that the suite is likely to enable shorter design time, higher quality of design and a shorter training time for novices. Furthermore, its results seem more predictable than manual and we expect that the novices that used the suite are more aware of the feasibility of their designs. These findings indicate that the suite does address the complexity of a collaboration process. Thus we can conclude that the suite meets the requirements as stated in section 3, but can be largely improved especially on the following aspects as suggested by the users.

A new suite should be more interactive, more “hyper linked”, concepts should be explained by a right mouse click or similar function, and it should deliver a full design, with format and introduction that is adjustable, can contain multiple modules and allows deviation from the suggestions of the suite.

It would be interesting to couple the suite to a GSS, for instance the next version of GroupSystems™, Cognito™. This would enable a facilitator to design the session directly in the GSS. This approach however also has a disadvantage; thinkLets as described in [20] have become technology independent. This flexibility is valuable; thinkLets can be used with several facilitation tools, making the facilitator technology independent.

The suite is focused on design and preparation. Good design is not good facilitation, the group dynamics and task aspects that are not predictable, or were not clear in the analysis can still force the facilitator to improvise during the meeting. Facilitation skills and experience are not replaceable with a tool. The suite might support easier memorization of thinkLets and therefore also during the session, support the facilitator. This is not evaluated and could be an interesting subject of further research. Also, the suite does not provide an experience of a GSS session, making it hard for complete novices to imagine what the thinkLets will do. A new version might include more scenarios and simulation of events.

In our opinion it is not useful to further test this suite, the evaluations indicate many improvements that are very likely to increase its ease of use and effectiveness. An improved version of the suite should be tested in a training situation with a larger timeframe and a larger test group to be able to confirm the results. However the evaluation does show that the suite is very likely to increase the success of novice facilitators.

The suite and observation of its users also gave rise to some questions about the thinkLet concept. The different thinkLets show overlap, some thinkLets are very similar, furthermore the categorizations are not unambiguous, and therefore the choice of a thinkLet is more complex than necessary. In Kolfschoten et al, [20] an attempt to improve the concept by decomposing it in more elementary components, is described. This research should be extended and evaluated in order to improve the concept and enable a more deliberated choice for a thinkLet.

Expert facilitators should together identify the best choice criteria for a facilitation technique or thinkLet. They should also identify a hierarchy in these choice criteria. We expect the choice criteria to involve aspects of the group and the task. Furthermore the choice criteria can only be applied to sufficiently decomposed techniques, or elementary thinkLet components as described above. A further elaboration on this is to create a suite that can build a thinkLet out of components, and can customize that thinkLet.

Acknowledgements

The authors would like to thank Jaco Appelman, Henk Sol, Bob Briggs, Gert-Jan de Vreede, Marielle den Hengst, Willem-Geert Phaff, Marcel van der Vos, Wilko van Oosten, all participants to the workshops, and reviewers for their support and participation in this project. Furthermore we thank EDUTEC for the financial support for the development of the suite.

References


