Recurring Patterns of Facilitation Interventions in GSS Sessions

Gwendolyn L. Kolfschoten1, Jaco H. Appelman1, Robert O. Briggs1,3, Gert-Jan de Vreede1,2

1Faculty of Technology, Policy and Management
Delft University of Technology
The Netherlands
G.L.Kolfschoten@student.tudelft.nl
J.H.Appelman@tbm.tudelft.nl

2College of Information Science & Technology
Department of Information Systems & Quantitative Analysis
University of Nebraska at Omaha
gdevreede@mail.unomaha.edu

3GroupSystems.com
1430 E. Fort Lowell Rd. #301
Tucson, AZ
bbriggs@groupsystems.com

Abstract
Now that GSS have been commercialized and are present in an increasing number of organizational settings, sustained use of GSS within organizations is an important subject of research. One of the aspects of GSS use that clearly poses challenges for organizations concerns GSS session design. ThinkLets are chunked facilitation techniques that can be used as building blocks for GSS sessions. In order to develop a library of useful thinkLets for organizations to use, this study researched patterns of thinkLets use in a large number of GSS sessions. We identified which thinkLets were used most and which patterns of thinkLets in terms of a fixed sequence emerged as ‘best practices’. By identifying such sequences, called modules, we can offer organizations powerful capsules of knowledge to wield GSS effectively. The results of our study represent a first version of a lexicon for the discipline of collaboration engineering with thinkLets and opens various avenues for research on making organizational collaboration more productive.

1. Introduction
In order to remain competitive, organizations must continuously seek to improve productivity. Productivity is the degree to which an organization employs its resources to attain goals with respect to creating value for its stakeholders, such as customers, stockholders, management, and workers. Extensive research shows that, under certain circumstances, teams that use Group Support Systems (GSS) can be significantly more productive than teams that do not. (See [7,8] for a compendium of GSS research.) However, GSS has been slow to attain widespread use in the workplace [1,5]. Therefore, many organizations are not deriving the benefit of the productivity gains that GSS may provide.

From a user perspective there is a fundamental difference between GSS and many other software technologies that inhibits the rapid spread of GSS. In other software, the primary purpose of the software is to allow a user to create and manipulate useful digital objects – data records, documents, images, sounds, and so on. These digital objects manifest in the user interface of the computer. While GSS also allow users to create digital objects, the primary purpose of a GSS is to support the creation of sustained predictable, repeatable, useful patterns of collaboration among people working together toward a goal [5]. These patterns do not manifest in the user interface of the computer; they manifest among users themselves. Indeed, in the current generation of GSS, there is nothing in the computer interface that informs a novice user about what patterns of collaboration could be created with a given configuration of the GSS tools. Given the variety of subtle and powerful changes one can make to the configuration of a GSS (e.g. anonymous or identified communication), the operation of GSS in the field can be quite complex. Although a user can learn the software controls for a GSS in a matter of hours or days, it can take as much as a year to learn how to configure and use the software to predictably create a variety of useful group dynamics.

To manage the complexity of GSS, many organizations depend on professional facilitators to wield GSS on behalf of other users [10,17]. This further inhibits the rapid spread of the technology, because facilitators are expensive, and because good facilitators tend to be promoted into management rather quickly, leaving nobody to wield the technology on behalf of the users [5].

In a variety of domains, people develop routines to cope with complexity, see e.g. [2,9,11]. In order that organizations might derive ongoing value from a GSS, and so perhaps to improve their productivity, researchers in the emerging field of collaboration engineering have begun to identify, develop, document, and validate codified facilitation techniques that together form a pattern language for group collaboration. These facilitation techniques are called thinkLets [5]. A thinkLet is a named, tightly scripted, process for creating a single repeatable predictable pattern of collaboration among people working together towards a goal. A thinkLet is defined in terms of the tools used, the configuration of these tools, and the script followed during a facilitation
Diverge – move from having fewer to having more concepts.
Converge – move from having many concepts to a focus on and understanding of fewer concepts deemed worthy of more attention.
Organize – move from less to more understanding of relationships among concepts.
Evaluate – move from less to more understanding of consequences for choices toward attaining group goals.
Build Consensus (also called Align Goals) – move from less to more agreement among a group of stakeholders; achieving more congruence between individual and group goals.

Although the concept of thinkLets has only recently been formalized, thinkLets have been with us for many years. Brainstorming [12], Nominal Group Technique [15] and Directed Brainstorming [13] are examples of divergence thinkLets that have been formalized and studied. Many other useful thinkLets may exist. If they are identified, documented, and studied, it might be possible to:

- Build a library of re-usable predictable facilitation techniques so that both novice and experienced meeting facilitators can benefit from validated successful interventions.
- Use them to design better team processes and increase the likelihood that organizations will remain productive.

The challenge therefore is to recognize which thinkLets have been successfully applied in GSS meetings. Further, it is important to realize that a thinkLet by itself usually does not constitute a complete collaboration process. A number of thinkLets are typically executed in sequence to help a group achieve its goals [5]. It is therefore also important to recognize collaboration process patterns, i.e. groups of thinkLets that appear to be applied together. Such knowledge would inform the definition of a core set of useful thinkLets and of modules of group processes that can be re-used.

Therefore, the focus of this paper is to investigate recurring patterns of group activities in GSS meetings. We examined the transcripts of more than 100 meetings that were supported with GroupSystems for Windows, a GSS developed at the University of Arizona and commercialized by GroupSystems.com. Through the detection of patterns of thinkLets we aim to investigate the use of thinkLets as a ‘meta-modeling or design language’ for group processes. The central research question addressed in this paper is: Are there repeated patterns of thinkLets that can be discerned in GSS sessions? Relevant sub-questions include:

- Can we find enough information, at tool and script level, to reconstruct thinkLets from past meetings?
- If we can reconstruct thinkLets in retrospect, can we detect recurring patterns of thinkLets?

The remainder of this paper is structured as follows. In the next section we describe the relevant concepts for our analysis. We present a model that defines all relevant concepts that we need to distinguish in a collaboration process. Section 3 gives an account of the way in which the research project was carried out. The results of our study are presented and discussed in Section 4. In the concluding section we summarize our main findings, describe the study’s limitations, implications, and propose directions for future research.

2. Background

Just as a brick does not constitute a building, so a thinkLet does not constitute a team process. A collaboration engineer must combine thinkLets with other components to create a complete group process design. Before we can examine such process designs we first have to define which components or elements of a process design have to be captured. Figure 1 illustrates the nature of and the relationships among these components.

The model depicted in figure 1 is a UML class diagram [3,14]. Each class is represented by a box that consists of three blocks. The upper block is the name of the class, the middle block lists the class’ attributes, and the lower block lists its operations. Relations between classes are represented by lines. There are different types of relations. The lines with the diamonds represent aggregation relations, meaning that the class where the diamonds are attached to consists of the other sub classes. A line with an open triangle represents an inheritance relationship (not present in figure 1). A line with a black triangle represents an association relationship. For example in figure 1, the thinkLet class may use the modifier class.

The thinkLet is the key class in figure 1. Each thinkLet has a name, and it creates a specific pattern of collaboration. ThinkLets are defined in terms of tool, configuration, and script [5]:

- **Tool** – The specific version of the specific hardware and software technology used. The Tool class has a name, version and configuration.
- **Configuration** – The specifics of how the tool is configured. In the UML diagram, the configuration is modeled as an attribute to tool.
- **Script** – The sequence of facilitation actions and instructions given to the group to execute the thinkLet. This is the only attribute of the Script class.

Small variations in a thinkLet are modeled with the Modifier class. Modifiers are named, repeatable variations that can be applied to thinkLets to produce a predictable change in the
thinkLets’ intended pattern of collaboration. It is useful to name and codify a modifier when it can be applied usefully to multiple thinkLets. Modifiers consist of only a change in the script. An example of a modifier is ‘OneMinuteMadness’: it changes a diverge thinkLet’s script to check early on in the process if the brainstorm instructions have to be redefined if the group’s contributions are inappropriate or of poor quality.

The next class of interest is the thinkLet transition class. Just as bricks must be joined with mortar to form a building, so thinkLets must be joined by transitions to make a collaboration process design. A thinkLet transition consists of all the actions required to shift from one thinkLet to the next. ThinkLet transitions include but may not be limited to the following:

- **Data transformations.** The output of one thinkLet may have to be transformed before it can serve as input to the next thinkLet.
- **Changes of participant orientation.** Participants may have to be alerted that the old activity has finished and a new one is beginning. This may also involve changing the attitude and expectations of the group towards the process.

It is useful to distinguish between thinkLets, Modifiers and thinkLet transitions. ThinkLets are basic activities that produce known patterns of interaction, the bricks in the building. Transitions are the changes from one thinkLet to another, the mortar. In this metaphor the modifier represents the role of paint. It does not change the function of the building but subtly influences the way in which the occupants will feel and experience it.

A number of thinkLets executed in a particular order can be defined with the Module class. A module is a compound thinkLet. It is a named, repeatable sequence of thinkLets and transitions that produces a known, useful result. The identifying characteristic of a module is that it works on a single dataset (such as participants’ ideas, contributions, or opinions). Every time the group starts working from a new set of data or starts creating a new data set, a new module starts. For example, in a collaboration process where a group has to decide on the location of a new office building, one module could address a divergence-convergence-evaluation sequence to identify a prioritized list of potential locations, whereas another module could address a divergence-convergence sequence to determine which criteria to use to select the actual location.

Modules can reduce the cognitive load for collaboration engineers and facilitators. Modules allow them to wield sophisticated and complex sets of facilitation interventions as a single concept. For our study it is important to identify modules, as each module in a collaboration process will produce a finished product, for example a prioritized list of options or an agreement. Individual thinkLets may also result in finished products (if they are the last thinkLet in a module), but usually thinkLets produce intermediate results that have to be further refined, elaborated, discussed etc. Therefore, by defining modules we can compare processes that are used to create similar end products.

Just as we need transitions between thinkLets, we also need transitions between modules. A **Module transition** prepares the move from one module to the next. It might be necessary to address a different subject in the next module. It might be necessary to prepare data for the next module. It might also be necessary to change the orientation and expectations of the participants, like in the thinkLet transition.

Finally, one or more modules together with module transitions make up the **Collaboration process.** A collaboration engineer creates, applies, and/or combines modules into a complete collaboration process design. The design is used to facilitate the GSS session. A collaboration process has a name, date, specific goal and a facilitator.

These concepts, now defined in detail, will inform the basis of the research on repeated sequences of thinkLets (modules) in GSS session design. The approach to this research is described in the next section.

### 3. Research Approach

Delft University of Technology in the Netherlands operates one of the longest-running GSS installations in the world, having opened in March of 1993. Over the past decade, Delft researchers and facilitators have conducted thousands of sessions for industry, government, and educational groups. The session transcripts have been collected and archived, sometimes together with other preparation materials such as preparation notes, meeting agenda, and introductory slideshows. These transcripts may reveal regularities in the process designs that could be codified as thinkLets and modules for future use.

In order to codify thinkLets from the archived transcripts, we had to identify the tools, configurations, and scripts that were used during the sessions. These session transcripts provided a number of clues. Each transcript included an electronically generated agenda. The agenda provided a list of electronic tools used by the group and the configuration of the tools at the time the team finished working with it. The transcripts also included the complete set of electronic contributions made to each tool. Sometimes the transcripts included additional written instructions to the groups to orient them about what was expected form them during each activity. Finally, each transcript included time and date stamps on each contribution. For some sessions we used additional archived information such as documents containing the design of the session, slideshows and interview notes.
We focused our analysis on the sessions that took place between 2000 and 2002, as there would still be a good chance to find the facilitators of these sessions to recall crucial information about the sessions’ design and proceedings. In this period, over 150 sessions were initially selected to be considered for inclusion in the study. This set was reduced by only retaining sessions that met the following criteria:

- **The session was not used for educational or demonstration purposes.** Sessions within (executive) educational programs mostly consist of small discussion exercises or survey’s to evaluate sessions. They do not adequately represent real organizational use of GSS. Furthermore, demonstration sessions only serve to illustrate the GSS’ functionality but do not address a real organizational goal.
- **The session consisted of at least one module.** A module consists of a series of activities that modify, extend or just use the same data set in the GSS. As soon as the participants created a new dataset this meant a new module started. A module can be seen as a sequence of activities.
- **The sequence of the activities in a module was clear.** We excluded sessions where no sequence of activities could be recovered. The sequence was not always clear from the agenda. Sometimes the agenda changed during the session while this was not captured as a deviation from an earlier agenda.
• The thinkLets were clear. The thinkLets could all be identified or constructed from the available information. Any uncertainties could be resolved through additional information gathering from the session’s facilitator.

Using these criteria, we identified the transcripts of 93 sessions that took place between 2000 and 2002. In these sessions, many of the facilitators had not deliberately used formal thinkLet scripts. A survey of 13 facilitators revealed that seven of them had used formal thinkLet scripts, and all had changed the scripts to adapt them to the needs of their group.

After selecting the sessions, we identified the thinkLets that must have been used by reconstructing them using the information available and through discussions with some facilitators. We identified 12 unique thinkLets, one of which had not been documented earlier. We named it “Attention!” Attention! is a divergence thinkLet that lets a group comment on listed items. A detailed description of Attention! is given in Appendix A.

4. Results and discussion

This section presents results of our analysis at several levels. We first show the frequency of use of thinkLets in general. Then we discuss the position which they generally occupy in a module. Finally, we identify and discuss various modules that are used frequently.

4.1 Frequency of thinkLets

Table 1 shows the frequency of use of each thinkLet that occurred more than 5 times in the set of meetings that we investigated. It also shows the position of a thinkLet in the sequence of activities in a module. The position of a thinkLet is discussed in more detail in the next subsection. For a detailed description of each thinkLet we refer to [4].

As can be seen in Table 1, the four most frequently used thinkLets were OnePage, LeafHopper, Attention! and StrawPoll. With OnePage all participants contribute simultaneously to the same electronic page or list. Each contribution appears instantly on the screens of all users. OnePage is similar to an electronic chat session. It is a divergence thinkLet that lets participants contribute to or comment on a list of brainstormed items by adding comments to it is a popular activity. A key reason for performing this thinkLet so often appears to be that it helps participants to create shared understanding during a follow-up activity. Participants share perceptions on certain issues – they make them explicit. In other words, it helps participants to add meaning to ideas. This is necessary before joint decision making and action taking can occur.

StrawPoll is an evaluation thinkLet that allows participants to cast an opinion on a set of issues using a variety of voting methods. It is used in many sessions because it is perceived to be fast and fun. It helps participants to attach values to the concepts proposed and gives participants a feeling of where the group as a whole stands or wants to go. Its popularity is likely to be also due to the fact that StrawPoll basically is the most generally applicable thinkLet to enable the evaluate pattern of collaboration.

Looking at Table 1, it can further be noted that thinkLets to create a Build Consensus pattern of collaboration were not frequently used. The large majority of sessions included in our sample did not aim to make a group decision or reach agreement on an issue. The sessions could be characterized as problem exploration or decision preparation.

Finally, feedback from facilitators and investigation of their meeting plans revealed that they quickly tend to develop habits as far as the choice of thinkLets goes. Once facilitators feel that a specific process combined with a specific thinkLet is successful, they are not very interested to further explore the possibilities the GSS offers. They do not feel the urge to discover and try out new thinkLets. A habit to use a certain thinkLet prevails. This obviously is risky, as the knowledge
Table 1. Frequency and position of thinkLets in modules.

<table>
<thead>
<tr>
<th>thinkLet name</th>
<th>#Occurrences</th>
<th>First%</th>
<th>Middle%</th>
<th>Last%</th>
<th>Standalone%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Diverge</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OnePage</td>
<td>66</td>
<td>92.4</td>
<td>0</td>
<td>0</td>
<td>7.6</td>
</tr>
<tr>
<td>LeafHopper</td>
<td>87</td>
<td>83.9</td>
<td>0</td>
<td>1.1</td>
<td>14.9</td>
</tr>
<tr>
<td>PlusMinusInteresting</td>
<td>37</td>
<td>35.1</td>
<td>0</td>
<td>64.9</td>
<td>0</td>
</tr>
<tr>
<td>Attention!</td>
<td>161</td>
<td>3.9</td>
<td>40.5</td>
<td>41.8</td>
<td>13.7</td>
</tr>
<tr>
<td>FreeBrainstorm</td>
<td>6</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td><strong>Converge</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FastFocus</td>
<td>36</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BroomWagon</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>12.5</td>
<td>62.5</td>
</tr>
<tr>
<td>CheckMark</td>
<td>47</td>
<td>0</td>
<td>19.1</td>
<td>66.0</td>
<td>14.9</td>
</tr>
<tr>
<td><strong>Organize</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PopcornSort</td>
<td>20</td>
<td>0</td>
<td>75.0</td>
<td>25.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Evaluate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrawPoll</td>
<td>75</td>
<td>4.0</td>
<td>20.0</td>
<td>37.3</td>
<td>38.7</td>
</tr>
<tr>
<td>BucketWalk</td>
<td>13</td>
<td>0</td>
<td>69.2</td>
<td>30.8</td>
<td>0</td>
</tr>
<tr>
<td>MultiCriteria</td>
<td>11</td>
<td>0</td>
<td>45.5</td>
<td>54.4</td>
<td>0</td>
</tr>
</tbody>
</table>

about a particular thinkLet appears to determine the agenda, and not the goal of the group activities.

4.2 Position of thinkLets in modules

Table 1 also depicts the position of each thinkLet in a module when it is part of a sequence. To inform the design of a session it is useful to know when a certain ThinkLet is used. This also gives us insight into how modules are build up.

It is not surprising that the majority of modules start with divergence thinkLets. They produce a set of items to be considered. StrawPoll and Attention! are used when there are already concepts to be considered at the beginning of a module, i.e. when there are ‘seeds’ in place. FastFocus is always used as an ‘in-between’ thinkLet. It is a very effective convergence thinkLet that helps the group to filter key ideas out of a larger set and build shared understanding of these ideas. Other popular ‘in-between’ thinkLets are the PopcornSort, used to organize a set of concepts into predefined categories, and the BucketWalk, used to check whether results of a PopcornSort or LeafHopper thinkLet are satisfactory.

At the tail of a module evaluation or divergence thinkLets dominate. Groups often prioritize the items or elaborate on items that are deemed important, to create further understanding or to weigh pro and cons before moving on to the next module, i.e. next part of the session. In our view, the goal of concluding a module like that is to decrease cognitive distance among the participants with respect to the concepts that were generated or addressed during earlier activities.

4.3 Modules

The result of our research to uncover modules is depicted in table 2. Only modules that occurred five times or more are taken into account. ThinkLets are coded as follows: the first letter indicates the pattern of collaboration the thinkLet aims to create, the other letters represent an abbreviation of the thinkLet name.

Modules usually consist of a sequence of 2 to 4 thinkLets. The most recurring sequence is a Leafhopper, PlusMinusInteresting or OnePage followed by Attention!. This module represents a double divergence that takes place on different levels. For example, in the first thinkLet categories and ideas within these categories are identified, while in the second thinkLet comments are added to the ideas. The popularity of this double divergence module can be explained by the nature of the problems addressed in the sessions that were included in the sample (see also section 4.1): highly complex, requiring attention to different aspects and perspectives.

Modules that consist of Leafhopper, Attention!, and an evaluation thinkLet also occur regularly. CheckMark and StrawPoll are popular evaluation thinkLets, and the combination of Attention! with these thinkLets is found even more. The value of this combination is that the group first elaborates on the items that are later voted on. In some cases the elaboration in Attention! specifically focused on the criteria that were use to vote on the list afterwards. In this way everyone in the group is informed about the different perceptions of an idea in the group, before they judge it. The
Table 2. Occurrence and composition of modules.

<table>
<thead>
<tr>
<th>#Occurrences</th>
<th>1st thinkLet</th>
<th>2nd thinkLet</th>
<th>3rd thinkLet</th>
<th>4th thinkLet</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>DAT</td>
<td>ECM</td>
<td></td>
<td></td>
<td>CFF = Converge - FastFocus</td>
</tr>
<tr>
<td>15</td>
<td>DAT</td>
<td>ESP</td>
<td></td>
<td></td>
<td>DAT = Diverge - Attention!</td>
</tr>
<tr>
<td>48</td>
<td>DLH</td>
<td>DAT</td>
<td></td>
<td></td>
<td>DLH = Diverge - LeafHopper</td>
</tr>
<tr>
<td>6</td>
<td>DLH</td>
<td>DAT</td>
<td>EBW</td>
<td></td>
<td>DOP = Diverge - OnePage</td>
</tr>
<tr>
<td>10</td>
<td>DLH</td>
<td>DAT</td>
<td>ECM</td>
<td></td>
<td>EBW = Evaluate - BucketWalk</td>
</tr>
<tr>
<td>10</td>
<td>DLH</td>
<td>DAT</td>
<td>ECM</td>
<td></td>
<td>ECM = Evaluate - CheckMark</td>
</tr>
<tr>
<td>7</td>
<td>DLH</td>
<td>EBW</td>
<td>ESP</td>
<td></td>
<td>ESP = Evaluate - StrawPoll</td>
</tr>
<tr>
<td>5</td>
<td>DLH</td>
<td>EBW</td>
<td>ECM</td>
<td>DAT</td>
<td>OPS = Organize - PopcornSort</td>
</tr>
<tr>
<td>32</td>
<td>DPMI</td>
<td>CFF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DPMI</td>
<td>CFF</td>
<td>ESP</td>
<td>ECM</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DPMI</td>
<td>DAT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>DOP</td>
<td>DAT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DOP</td>
<td>DAT</td>
<td>ECM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DOP</td>
<td>DAT</td>
<td>ESP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DOP</td>
<td>OPS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

same principle is found in modules that start with the PlusMinusInteresting thinkLet. The exploration of given concepts with PlusMinusInteresting, followed by the summary with the FastFocus, yields a very detailed exploration of opinions on concepts. This informs the evaluation in the end of the module.

When a module consists of a LeafHopper, (Attention!) and BucketWalk sequence, the group brainstorm in categories (and elaborates) on the ideas in it. Afterwards the group evaluates the brainstorm on redundancy and categorization. Such a module results in a clean categorization of items and is used for instance to explore the strategic directions of an organization. This module might very well result in an end product but we also found some modules that used the result to evaluate it and elaborate on a selection of the categorization.

The last module that often occurred is the combination of OnePage and PopcornSort. This combination actually is a reverse LeafHopper. Instead of brainstorming in categories, the group first explores an issue or situation by brainstorming from scratch so that they cannot be steered in their contributions. Next, the group categorizes their ideas.

5. Conclusion

ThinkLets are building blocks for organizational collaboration using GSS. ThinkLets are independent, reusable and replaceable. They are elementary and customizable with modifiers. Through transitions we can define their interfaces. Our study has shown that it is possible to model and analyze GSS meeting processes by perceiving them as a sequence of thinkLets.

In 93 sessions we found and described 15 unique modules, recurring patterns (sequences) of thinkLets that occurred 5 times or more. These modules offer an unique opportunity to explore in further detail and codify ‘best practices’ used by experienced GSS facilitators. Modules may boost efficient GSS session design and execution as they represent high level building blocks that accurately describe tool choice and configuration, and facilitation script. In other words, modules not only support inexperienced facilitators by providing powerful building blocks that can be drawn from a library but they also give insight in decisions in design that are found on a higher level than the level of the individual thinkLet.

Furthermore, the modules we found have the potential to inspire the design of new modules. To this end, thinkLets and modules form the foundation for designing recurring collaboration processes. This is done by collaboration engineers who combine thinkLets, modifiers, transitions and modules into a complete collaboration process design. The collaboration process components as defined in this study and presented in figure 1 constitute the beginning of a lexicon for collaboration engineers.

5.1 Limitations

There are a number of limitations that have to be taken into account when interpreting the results of this study. First, we have reviewed more than 150 sessions and analyzed 93 of them. Some of these sessions were executed before the thinkLet concept was introduced to the facilitators of these sessions. The reconstruction was therefore not always straightforward. We experienced some difficulty in the reconstruction of some scripts, which left some room for ambiguity. We tried to resolve any issues through focused interviews with the facilitators involved. Second,
facilitators develop their own routines. Facilitators that were responsible for the sessions included in the study learned most of their routines in tool use through word-of-mouth and trial and error in sessions. Since most facilitators came from one department, this may also have caused a certain style to develop over time that is idiosyncratic to the department. These routines and style may not reflect facilitation practices around the world. Yet, the department involved is specialized in designing collaboration processes for problem solving and action formulation, so the identified modules are grounded in a broad experience base in this area. Finally, we did not define and investigate transitions yet. It turned out to be too hard to identify transitions in retrospect as many transitions are not explicitly recorded. They are just executed during the session.

5.2 Implications and future research directions

Our research showed that the criteria to choose a particular thinkLet are not only based on the nature of the thinkLet itself, but also on its place in a sequence in a module. The decision which thinkLet to use is influenced both by the preceding as well as the subsequent thinkLets in the meeting process. For example, a FreeBrainstorm thinkLet will result in broad set of contributions, likely containing a lot of redundancy. This may force the facilitator to choose a particular convergence thinkLet such as FastFocus to remove redundancy and enhance shared understanding. Yet if the facilitator intends to assign subsets of the brainstorm results addressing particular aspects to smaller teams (DealersChoice), he may decide to use a PopcornSort after the FreeBrainstorm. With a PopcornSort the participants can organize the brainstorm results into a set of aspect categories. In order to further support novice and experienced facilitators to make informed choices about which thinkLets to use, more research is required to determine that nature and content of the selection criteria.

Another important area of research concerns the nature and selection criteria for modules of thinkLets. Module needs to define not only the combination of the thinkLets but also the transitions between them. So far, thinkLet transitions have not been researched yet they are expected to have a key role in the design of GSS sessions. Interesting issues in this respect concern the exact components of a transition and the potential to automate transitions between thinkLets.

Regarding modules we need to investigate which criteria underlie the selection and composition of modules in terms of thinkLets and thinkLet transitions. Without an understanding of these criteria we can only hope to uncover modules through detailed analysis of past GSS sessions; it will not be possible to consciously design modules, nor evaluate their effectiveness.

Finally, the lexicon of collaboration engineering concepts in terms of thinkLets, modifier, thinkLet transitions, modules, and module transitions offer a new way of looking at collaboration technologies. It will be interesting to prototype and investigate the potential of GSS that allow facilitators to design and re-use these collaboration engineering concepts directly instead of just selecting GSS functions and their configuration.

References

Appendix A. Attention!

A.1 Overview

With Attention! participants elaborate on their ideas, react on ideas of others, illuminate specific aspects and indicate which ideas they don't understand. Choose this thinkLet:

- When you want the participants to understand all ideas of a brainstorm.
- When you want the participants to elaborate on a specific aspect of the brainstormed items.

Do not choose this thinkLet:

- When the brainstorm concerns a non-complex topic, and the ideas are clear.

Input:

- A set of brainstormed ideas.

Output:

- A set of brainstormed ideas understood and clear for all group members.
- A better, more detailed documentation of the group memory.

A.2 How to Use Attention!

Setup

- Post the brainstorm items in a list-building tool.
- Allow participants to add comments; display comment numbers.
- You can actually do this halfway during another divergence activity, allowing both further brainstorming and commenting on the items that have already been generated.

Steps

1. If necessary point out specific aspect that need to be highlighted in the comments.
2. Say this
   a. Read what others have written (so far).
   b. If any item is unclear, please double click the item and point out what is unclear.
   c. If you feel your own items are not clear enough or misunderstood, please elaborate so that everyone will understand.
   d. Please elaborate on your own ideas on the specific aspect I just described.
   e. Check your own items on questions and if you can, please answer them.
   f. Give reactions on the items that are already in the list and add on the items of your expertise.
   g. Refer to comment numbers if you react on a comment instead of a list item.
3. Read through the comments and write the numbers of the items with questions on a whiteboard.
4. Put attention to the questions that were not answered yet.

A.3 Insights on Attention!

The Attention! thinkLet is a transition thinkLet that moves the group from having a large list of raw ideas to having a list with added detail that addresses particular aspects of these ideas. It results in enhanced understanding of the ideas. Aspects addressed during Attention! could for instance concern cost indication or participant emotions towards ideas. It is particularly useful when you want participants to evaluate a set of ideas with respect to a number of criteria. With Attention! you can let them first discuss each idea from the perspective of these criteria. Attention! provokes the participants to think about each idea from a particular perspective. Therefore, it makes a subsequent thinkLet for converging, organizing, evaluating or building consensus on the ideas easier. Attention! also improves the quality of the brainstorm report as it reduces the risk that people may interpret the results of the session in different ways.

You can combine this thinkLet with other divergence thinkLets such as OnePage, Leafhopper, BranchBuilder, ComparativeBrainstorm, or DealersChoice. Merging Attention! with another thinkLet may however result in information overload. Especially when creating a structure as in Leafhopper and BranchBuilder, adding one more level of detail can be too much. Therefore, when the structure of ideas is very large, consider using a convergence thinkLet first to reduce it.
A.4 What's in a name?

In the Army the officer will yell “Attention!” to the platoon and all soldiers will jump up and look the same direction. The Attention! thinkLet aligns the group in similar fashion making sure everyone addresses and contemplates the ideas in a list structure.