Facilitation Tool – A Tool to Assist Facilitators Managing Group Decision Support Systems

Pedro Antunes
Tania Ho

Department of Informatic Engineering,
IST, Technical University of Lisboa,
Av. Rovisco Pais, 1049-001 Lisboa, Portugal

Abstract

This paper describes the Facilitation Tool, a tool built to untangle two problems that we find in current Group Decision Support Systems. If we want to allow inexperienced facilitators manage such systems with success, then a planning functionality more detailed than the currently available is necessary. We also found very limited support to remote facilitation. Several notions concerning decision-making and facilitation, which were the basis for our design, are described. The software architecture and some components of the Facilitation Tool are explained. Finally, results obtained from an experiment with the planning functionality are presented.

Keywords: Electronic Facilitator, Group Decision Support Systems.

1   Introduction

Group facilitation is a process in which a person who is acceptable to all members of the group intervenes to help improving the way it identifies and solves problems and makes decision [24]. Facilitation is one of the several third party processes studied in organisational behaviour [15]: mediation (conciliate interactions among disputants), arbitration (resolve disputes while giving equal opportunities), inquisition (stronger than arbitration) and process consultation (facilitate problem solving).

There is an increasing presence of GDSS in organisations, augmenting the demand for people trained to assist GDSS usage, i.e. electronic facilitators. Organisations are training their managers to facilitate meetings, an attempt to avoid prohibitively costs, but the transition from manager to facilitator is not considered easy [26] and thus the electronic facilitator is still a scarce organisational resource. Hence, computational support is indispensable to assist inexperienced facilitators managing GDSS with success.

Cost reductions also increase the preference for remote meetings. In that situation, facilitators must rely on computer-mediated communication to intervene in the group, which, depending on the available channels, requires additional effort and reduces the variety of interventions. Facilitation support must be designed to uphold remote facilitation roles; a functionality that goes beyond the most commonly supported chauffeur-style interventions, e.g. shifting tasks.

The paper reports our efforts in the development of a tool dedicated to assist facilitators in the task of managing GDSS. Emphasis has been put on the two aspects identified above: (1) the pre-meeting design; and (2) support to facilitation interventions in remote sessions. The paper is organised in the following way. First, we overview related work concerning the support to electronic facilitation. Next, we present the set of requirements, followed by design and implementation details. Finally, we present results of the tool evaluation.

2   Related Work

An electronic facilitator executes four functions [23]: (1) provides technical support by initiating and terminating specific software tools[1]; (2) chairs the meeting, maintaining and updating the agenda and intervening when necessary[2]; (3) assists in agenda planning; and (4) provides organisational continuity, setting rules and maintaining an organisational repository. Thus, the facilitation process exists before, during and after meetings. We will address pre-meeting support and technical support during meetings. Post-meeting support is discussed elsewhere [5].

2.1   Pre-Meeting Support

It has been suggested that the importance of the electronic facilitator is that it forces the group to plan or design the meeting [4]. Niederman et al. [19] found out that a high number of high-experienced facilitators (25%) regard having a good agenda as a critical factor of meeting success. It is also interesting to note that planning and

---

[1] Designated chauffeur [12]. The chauffeur uses the technology guided by the group and does not affect the process.
[2] With the objective of improving productivity and quality of results [6][16].

problem-solving skills were mentioned by 14% of the facilitators as critical factors to their own success. A sequel to this study [20] shows that facilitators find agendas to have significant impact on meeting outcomes (3.3 to 4.4 on a scale of 1 to 5), particularly in what concerns the quality of outcomes.

The agenda building core functionality is extremely simple: assemble a list of tools to be orderly invoked during the meeting[3]. SAMM [7] provides an agenda with the possibility of defining sub-agendas, enter, view, modify and delete topics. The GroupSystems' [21][22] agenda tool allows the facilitator to organise multiple meetings in folders and, for each meeting, define the sequence of techniques to invoke (e.g. electronic brainstorming, categorizer, vote and so forth). The agenda also provides some additional facilitation aids, with elements such as *introduction, lunch* and *coffee break*. Using the agenda, the facilitator can also name and describe agenda topics, define time limits and select participants. The Meeting Works GDSS also provides an agenda planner, where topics and tasks can be organised in a list. Meeting Works separates functionally the agenda planner and the meeting chauffeur. The later is responsible for matching agenda tasks with the techniques supported by the system.

Other non-electronic systems can also be identified in this category: Graphic facilitation [26], a set of symbols, pictographs and ideographs, to visually organise meetings; and Facilicom [31], a kit with components which can be placed in chalkboards or flip charts during meetings.

One characteristic that is clear in the functionality described above is that facilitators must rely on past experience to assemble a suitable list of tool. No kind of support is given to inexperienced facilitators.

Meeting Works and the latest version of GroupSystems Workgroup Edition supplement the agenda building functionality with pre-defined agendas, i.e. collections of representative patterns. This new functionality consolidates reusability and adaptation in meeting design. For instance, Meeting Works has pre-defined agendas for group development, checkpointing, developing a new project and performing strategic planning. However one should consider that, on the one hand, such pre-defined agendas focus on very

specific processes and thus have low reuse potential, and on the other hand, they do not cover the whole spectrum of decision-making situations faced by facilitators. The configuration and adaptation of pre-defined agendas to different processes requires, once again, expertise on facilitation, which means that pre-defined agendas do not represent a solution to the support for inexperienced facilitators.

Expert Session Planner [1] is a prototype expert system designed to support pre-meeting planning of decision processes. According to the authors, various models of task characteristics, nature of the problem and other characteristics such as need for consensus are included in the ESP. Based on these models, the ESP makes tool recommendations to the facilitator. The Matcher [3], a system designed to interconnect workflow and GDSS systems, provides a similar functionality. The Matcher is responsible for the identification of situations where workflow systems cannot progress and informal decisions must be taken. It has a set of models that allow to select the type of decision-making process most adequate to the situation. With this functionality, facilitators just have to provide answers to a collection of generic questions (e.g. need for consensus) and obtain an agenda that best fits the problem context.

Our concern is that, since the reasoning behind automatic agenda building is not perceptible, inexperienced facilitators are unable to develop their skills. This view leaves an open door for further improving pre-meeting support. The alternative explored by this work regains the notion of pre-defined agendas, but attempts to specify a process with support to generic decision-making patterns capable to aid an inexperienced facilitator making options in a wide range of situations.

2.2 *Support During Meetings*

Clawson et al. [4] report an inquiry with 50 facilitators, which revealed 16 generic critical facilitative behaviours, being the most frequently cited by order of importance: build rapport and relationships; plan and design meetings; and direct and manage meetings. 71% of the identified facilitators' actions correspond to interventions during meetings.

Viller [27] identifies 5 generic group syndromes that facilitators must confront: multi-headed beast, feuding factions, dominant species, recycling and sleeping meeting. Two methods are proposed to intervene in such situations: interpretation (change group focus to the process) and direct action (directly manipulate the process).

The study from Viller [27] also raises the problem of remote facilitation. In that situation, facilitators

---

[3] We do not address support functionalities such as the ones supported by meeting scheduling systems (e.g. Lotus Notes and Ms. Outlook): calendar, participant rooster, means to visualise others' agendas, invite meeting participants or automatically schedule sessions.

must rely on computer-mediated communication to intervene in the group, which may reduce the range of possible interventions.

Shift task is the most useful GroupSystems' functionality supported in a remote facilitation context. *Shift task* transfers data from one technique to another. SAMM provides a set of basic utilities during meetings, such as log files and meeting minutes. These features are available to all participants, since SAMM is user-driven[4]. The Distributed Facilitation System [9] is a research prototype that specifically addresses several facilitation functions during remote meetings, classified as recording (transcripts, snapshots and summary), monitoring and process (start/stop). Other functions supported include start-up (enrol participants) and wind-up (tracking accomplishments).

The interventions described above fall in a category that may be classified as chauffeur-style interventions, dedicated to manage the technology bot not the process.

GroupSystems has an interesting *opinion meter* that falls in a different category. The *opinion meter* is a lightweight voting mechanism that allows fast decisions any time during meetings. On the same vein, Expert Session Facilitator [2], a prototype expert system designed to support electronic facilitators during meetings, monitors the number of comments from each participant and sends reminders to contribute more. It also detects drop of frequency and contents of contributions, indicating when to terminate the session. Group Work Environment [18] has also a consensus component dedicated to assist facilitators in analysing group status through preference elicitation and analysis of a set of alternatives. Preference data is analysed using two metrics, participants' consensus and agreement. Further examples are the Consensus Response Keypad (question/answer, inquiries, rankings and other games [29]) and Council2 (gathers ideas fast and at any moment during a meeting [30]).

The above interventions fall in a new category that aims at harvesting preferences and participation with the intention of intervening in the decision-making process.

Considering the 16 behaviours identified by Clawson et al. [4], we must conclude that there is still limited support to electronic facilitation of remote GDSS sessions. Several missing interventions are explored in this paper: steering the group, managing conflicts or keeping the group focussed.

---

[4] Any participant may assume at any moment the role of facilitator [10][28].

3   Facilitation Tool – Requirements

We developed the Facilitation Tool (FT) with the objective of assisting facilitators before and during GDSS sessions. The tool had to accomplish the following functional requirements:

- Support pre-meeting planning adequate to inexperienced facilitators. In particular, the tool had to explicitly provide a model of the decision-making process.
- Support remote facilitation, with provision of mechanisms for remote interventions in meetings.
- And be independent but easily integrated with GDSS, for instance GroupSystems and Meeting Works.

The final requirement came out to avoid developing yet another GDSS and contributed most to the actual architecture of the developed tool.

3.1   Definition of a Model

This section elaborates the first one of the requirements specified above: provision of a model of the decision-making process. Decision-making can be viewed as a system, with a vision, mission, operational objective, and also a project and improvement plan. The project may be briefly specified as "to select that one of the strategies which is followed by the preferred set of consequences" [25]. This kind of project may be decomposed in a set of tasks, or process. We start by briefly presenting several process models and then justifying our selection.

Simon [25] describes a rational model with three steps: listing all alternative strategies; determination of all consequences that follow upon each strategy; and comparative evaluation. McGrath [17] defines a task typology classifying group tasks in four clusters: generate, choose, negotiate and execute. Each cluster is further subdivided according to more specific cooperation/conflict and conceptual/behavioural characteristics. Schwarz [24] presents a nine-step problem-solving model: define the problem; establish criteria for evaluating solutions; identify root causes; generate alternative solutions; evaluate alternative solutions; select the best solution; develop an action plan; implement the action plan; and evaluate outcomes and the process. Johansen et al. [13] present a model with seven steps: orientation, trust building, goal/role clarification, commitment, implementation, high performance and renewal.

Kaner [14] proposes a more complex model, which will be briefly detailed. A problem may be divided in a set of more specific issues, each one requiring a decision-making process. Each process develops

according to a sequence of different zones. There are four different zones, which come in the following temporal order: divergent (search for information); groan (discuss issues); convergent (attempt to reduce the number of solutions); and closure (select one solution by consensus or voting). Each zone can consist of one or more strategies (decision patterns) for handling the issue. For instance, *exploring the territory*, *searching for alternatives* or *discussing difficult issues* are different strategies defined for the divergent zone. Finally, a strategy can consist of one or more activities. As an example, we find in the *explore the territory* strategy a sequence of activities characterised as *who, what, when, where and how* (identify who is involved, what must be done and so forth).

Hwang and Lin [11] divide the decision making process in four phases: extraction, exploration, selection and execution. The authors also propose an interesting classification of problem-solving purposive methods (computational or not) to implement the above phases: creative confrontation; polling of experts/participation; systematic structuring; simulation; and implementing and controlling.

Although all of the previously presented models were reasonable to use, we adopted the Kaner's model considering the following reasons:

- It is the most detailed one. In particular we found compelling the separation of concerns in multiple levels: issues, zones, strategies and activities.
- Each different strategy mentioned by Kaner is a very expressive and reusable pattern for handling a problem, increasing the opportunity for developing facilitators' skills.
- The activities are independent from tools particular to any GDSS. This is a good design practice, well known in software development, where implementation options are delayed as much as possible in the product life cycle.

The Kaner's model was extended in order to embrace two new levels of abstraction: task level and tool level. Both levels are intended to smoothly approximate the high-level decision-making design towards the actual process instantiation. The task level borrows the Hwang and Lin's characterisation of computational methods in five categories, but excludes simulation, given that such task is not addresses by the GDSS cited in this paper. The tool level directly maps tasks into GDSS tools such as brainstorming, topic commenter, categorizer, etc. This final level is the only one dependent from the particular GDSS used, while the other levels are qualified for reuse.

In Figure 1 we present a table descriptive of the decision-making process model specified for FT.

3.2 Remote Facilitation

This section elaborates the remote facilitation requirement. Based on the types of facilitator's interventions described in [24] and [14], we elected for implementation a subset that we believe to be most adapted to electronic meetings.

Interaction techniques, basically intended to steer and focus the group:

- Paraphrasing – The facilitator repeats what the participant said using own words.
- Mirroring – The facilitator repeats the participant's exact words.
- Balancing – The facilitator attempts to make silent participants to speak.
- Drawing people out – The facilitator asks one participant for more information.
- Encouraging – The facilitator encourages others to speak**.**

Analyse and understand techniques, dedicated to obtain feedback information from participants:

- Listening for common ground – This technique develops in four steps: (1) the facilitator says that is going to summarise; (2) makes a summary of divergences; (3) makes a summary of common views; and (4) asks if the participants agree with the list.
- Tracking – This techniques develops in three steps: (1) the facilitator says that is going to make a summary of the meeting; (2) identifies discussed topics; and (3) asks if the participants agree with the list.

Control techniques, to moderate conflicting or chaotic situations:

- Stacking – The facilitator organises and schedules the participants' interventions: (1) asks for anyone to speak; (2) makes a list of candidates; (3) schedules candidates; and (4) asks if anyone else want to speak.

Given that these interventions impose a burden to the facilitator, we developed a set of standard messages, which the facilitator may select for automatic delivery. These messages are presented in Figure 2.

Zone	Strategy	Activity	Task[5]	Tool[6]	
				GS	MW
Divergent	Explore territory	Say point of view	CC	TC	GEN
		Specify requirements	2 x CC / SS	TC / CAT	GEN / ORG
		Who, what, when, where, how?	5 x CC	TC	GEN
		Facts and opinions	2 x CC / CC	TC	GEN
		Initial positions	CC	BST	GEN
		Perspectives not represented	CC	BST	GEN
	Search for alternatives	Brainstorming	CC	BST	GEN
		Analogies	CC	BST	GEN
	Discuss difficult issues	Something not said?	CC	TC / CAT	GEN
		How does it affect me?	CC	TC / CAT	GEN
		3 complains	CC	TC / CAT	GEN
Groan	Create shared context	Learn others' perspectives	SS	CAT	ORG
		If I where in your place…	SS	CAT	ORG
		Solutions and needs	SS	GO	ORG
		Alternative futures	IC	GO	CROSS
Convergent	Reinforce good ideas	Clarify criteria	SS	GO	ORG
		Risks and consequences	SS	GO	ORG
		Who else needs to evaluate?	SS	GO	ORG
		Who does what when ?	SS	GO	ORG
	Explore principles	Case studies	IC	TC	CROSS
	Creative re-contextualisation	What cannot be changed?	IC	TC	CROSS
		Keywords	IC	TC	CROSS
		Revert assumptions	IC	TC	CROSS
		Remove restrictions	IC	TC	CROSS
		Catastrophising	IC	TC	CROSS
Closure	Voting	Doyle and Straus Fallback	POLL	VOT	EVAL
		Vote to Vote	POLL	VOT	EVAL
		Meta-Decision	POLL	VOT	EVAL

Figure 1 - Decision-making process model

Type	Intervention	Start text	End text	Receiver
Steer and focus	Paraphrasing	"I think you are saying" _____ "Let me see if I understood" _____ "It sounds like you are saying" _____	"Did I understood correctly?" "Is that right?" "Did I get it?"	One
	Mirroring	"You said" _____		One
	Balancing	"Does anyone else have a different opinion?" "Are there any different perspectives?" "Who else has an idea?" "What do others think?"		All
	Drawing people out	"Can you say more about that?" "What do you exactly mean by that?"		One
	Encouraging	"Does anyone have more ideas on that topic?" "Does anyone have problems with this issue?" "Are there any different views?" "Lets hear from someone who hasn't spoken"		All
Analyse and understand	Listening for common ground	"Wait, I'm going to summarise… You have presented the following differences:" _____ "You have presented the following common views:" _____	"Have I got it right?"	All
	Tracking	"Wait, I'm going to summarise the meeting… You have discussed the following topics:" _____	"Do you agree with this summary?"	All
Control	Stacking	"Do you wish to speak?" "Does anyone else want to speak?"		All

Figure 2 – Standard messages delivered to participants

---

4 Facilitation Tool – Implementation

4.1 Platform and System Architecture

The FT has a client-server architecture, consisting of the Facilitation Server and Java Applets (clients) which can be downloaded from a WWW home page using a standard browser. There are two types of clients: the facilitator and participants of group activities. The server mediates all communication between facilitator and participants.
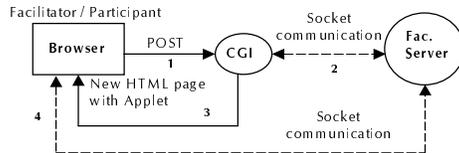
Figure 3 – Client initialisation

The initialisation of a new client is done in four steps, as illustrated in Figure 3. The new client accesses the FT main web page. After submitting a username and password (1), a CGI runs and establishes communication with the Facilitation Server (2). The CGI generates a new HTML page and sends it back to the client (3). The HTML page contains a Java Applet, which establishes communication with the Facilitation Server (4). Communication is done through TCP/IP sockets.

After initialisation, each client is running two threads, Applet and ClientProtocol, the former handling user interactions and the latter handling messaging between facilitator and participants, mediated by the Facilitation Server. The server executes several threads: a main thread, one thread serving new client connections and one ServerProtocol for each new client to communicate with the companion ClientProtocol.
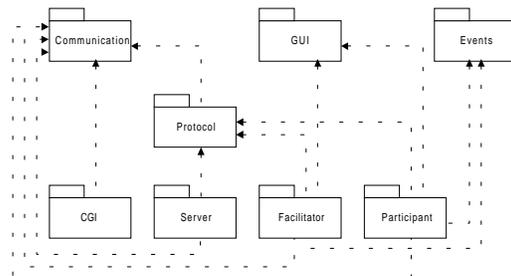
4.2 Software Architecture and System

Figure 4 – FT software architecture

The FT is implemented with eight software packages (Figure 4): Communication, Events, GUI, Protocol, Server, Facilitator, Participant and CGI. Of these, we will describe the Facilitator and Protocol packages in more detail. We will also partially describe the GUI package with the help of screen dumps taken from the FT.

4.2.1 Facilitator
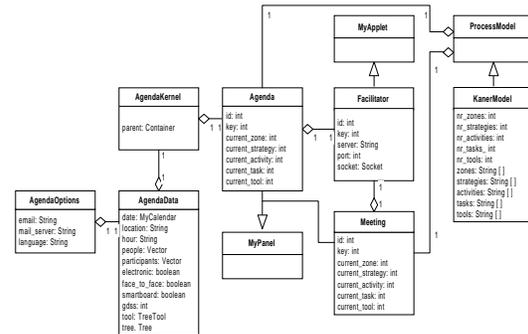
Figure 5 –The Facilitator package

The classes that comprise this package are presented in Figure 5. Of these, we highlight: the Agenda, with hierarchical definition of issues, each one having a name, description, expected results and optional comments; the ProcessModel, an abstract class that specifies the decision-making process; and the Meeting class, which allows visualising issues, registering their accomplishment and starting the facilitation techniques. The facilitation techniques are handled by the Protocol package described in the next section.

4.2.2 Protocol
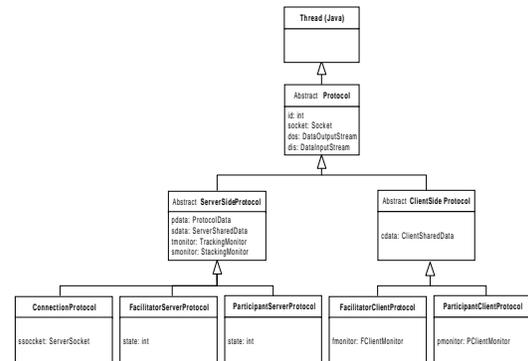
Figure 6 –The Protocol package

The Protocol package (Figure 6) has a set of classes that define the behaviour of the ClientProtocol and ServerProtocol threads running in the system. The package also includes classes dedicated to monitoring (Java synchronisation) and data sharing (between threads running in the server), not shown in Figure 6.

The remote facilitation techniques were organised according to three different protocols:

- No rounds – The facilitator only sends one message to participants (one or all of them). It includes balancing, drawing people out, encouraging, paraphrasing and mirroring.

- Two rounds – The facilitator sends a message to the group, waits for their responses and reports back to the group. This category includes listening for common ground and tracking.
- Multiple rounds – Similar to the previous situation but the facilitator reports back to selected individuals one by one. It includes stacking intervention.

To illustrate how these protocols work, Figure 7 shows the two-round situation. The facilitator starts the technique by pressing the START button in the Applet. The Applet notifies the Protocol package that a technique started (SetTechnique). Then, the FacilitatorClientProtocol sends a START message to the server. The FacilitatorServerProtocol checks for connected participants and responds with OK or ERROR (no participants). The FacilitatorClient Protocol sends then an EXECUTE TECHNIQUE event to the Applet. The facilitator selects one question to deliver to the group and presses the OK button in the Applet. The question goes through the FacilitatorClient, FacilitatorServer, Participant Client and participant Applets. The Facilitator ClientProtocol collects responses from all participants and delivers the results to the facilitator and also to the participants. The Facilitator ClientProtocol finishes the protocol by sending a STOP message to the Applet and server. The server transmits the STOP message to the participants.

Figure 7 – Two-round protocol

4.2.3   GUI

The GUI package contains a large set of classes that support interaction with users. Due to the limited space, we only present screen dumps that illustrate process design, process management during meetings and the encouraging and drawing people out techniques.

Figure 8 shows how the facilitator designs decision-making processes aided by the model. At the top left, the facilitator can break down the problem hierarchically into issues. At the centre of the window, the facilitator can select zones and corresponding strategies. To the right of the window, the facilitator finds a table for the selection of activities, tasks and tools. Finally, the bottom left window shows the process steps assembled by the facilitator, in the form of summary lines with the selected zones, tasks and tools. The facilitator may add or delete summary lines. The facilitator can also add contextual data to the summary lines.

Note that at this pre-meeting stage the facilitator is the only client connected to the Facilitation Server.

Figure 8 – Process design window

Several GUI details concerning participant rooster, meeting initialisation and management of connected participants are not illustrated in this section.

Figure 9 shows the main window which facilitators interact with after meeting initialisation. It displays the hierarchy of issues, process specification and status. The facilitator must tick the check boxes corresponding to finished tasks. From this window, the facilitator may start the facilitation techniques.

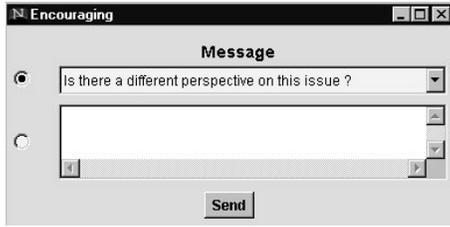Figure 9 – FT window during meetings

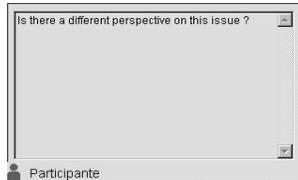Figure 10 - Facilitator's window for the encouraging technique

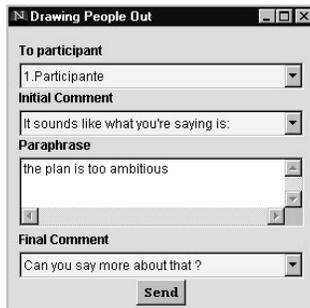Figure 11 - Participants' window for the encouraging technique

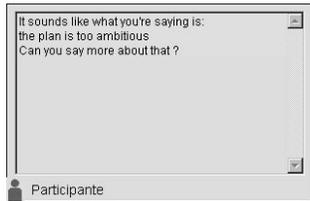Figure 12 – Facilitator's window for the drawing people out technique

Figure 13 – Participants' window for the drawing people out technique

Figures 10 to 13 present the encouraging and drawing people out techniques. The facilitator's window allows selecting any appropriate pre-defined messages and participants, when applicable. The window displayed to the chosen participant(s) shows these pre-defined messages.

5   Evaluation

So far, we have made a preliminary evaluation of the pre-meeting functionality of FT. The experiment, which confronted agendas generated with FT and GroupSystems, was set up as follows.

Four facilitators with low/moderate skills in electronic facilitation were selected to participate in the experiment. The experiment was done face-to-face at our electronic decision room, but the participants were requested to execute the tasks individual and silently. Each facilitator was requested to design agendas for the two following problems:

- Problem #1 – A car company must decide either to launch a new model or wait for the design team to introduce a new feature, which will make that model ahead of the market.

- Problem #2 – A training course is taking more than the 9 months expected. The problem is how to avoid delays without reducing quality.

Two facilitators were randomly selected to design problem #1 with GroupSystems' agenda and #2 with the FT, while the two other facilitators were requested to design the problem #1 with the FT and #2 with GroupSystems' agenda. With this procedure we obtained two alternative designs for two different problems.

A senior facilitator (with more than 30 years of experience on the subject) was requested to rank each agenda in a scale of 5 points (1 - lowest and 5 - highest result) using four criteria: structure, logical sequence clarity and efficiency of the agendas.

	GroupSystems		FT	
Problem #1	BST, CAT, VOT, CAT, VOT	CAT, CAT, CAT, VOT	TC, TC, TC, CAT, GO, TC, TC, TC, CAT, GO, GO, GO, VOT	BST, TC, TC, CAT, CAT, TC, CAT, CAT
Structure	2	3	2	3
Logical sequence	2	4	3	4
Clarity	1	4	3	3
Efficiency	1	4	2	2

	FT		GroupSystems	
Problem #2	BST, VOT, TC, CAT, VOT	TC, TC, BST, GO, CAT, GO	BST, CAT, VOT	BST, CAT, CAT, BST, VOT
Structure	3	2	4	2
Logical sequence	3	3	4	2
Clarity	4	2	4	2
Efficiency	2	2	4	2

Figure 14 - Raw results

The obtained results are summarised in Figures 14 and 15[7]. We added an explaining variable, number of tasks, to the average results shown in Figure 15.

---

[7] In Figure 14 we only present the layouts of the designed agendas but the senior facilitator had access to the design rationale behind each sequence of tools selected by the participants.

	GS	FT
Number of tasks	4.25	8
Structure	2.75	2.5
Logical sequence	3.0	3.25
Clarity	2.75	3.0
Efficiency	2.75	2.0

Figure 15 - Average results

From the results above, and considering their limited number, we may draw some preliminary conclusions.

First of all, the FT tends to generate a greater number of tasks (8) than the GroupSystems' agenda tool does (4.25). According to the senior facilitator, a greater number of tasks reduces the efficiency of the process (GroupSystems' agenda scores 2.75 and FT scores 2.0). As a fact, the best ranked agenda is the simplest one, comprising only three tasks (brainstorming, categorising and voting) obtained with GroupSystems' agenda. The senior facilitator also considered that a simpler agenda has better structure, and thus the GroupSystems' agenda again obtains better scores (2.75 versus 2.5).

On the other hand, the FT received best scores in the logical sequence (3.25 versus 3.0) and clarity (3.0 versus 2.75) criteria. In the end, both the GroupSystems' agenda and FT were better in two criteria.

It is also interesting to note that two facilitators had better results using the GroupSystems' agenda and two facilitators had better results using the FT.

Overall, we cannot conclude that one approach is better that the other. As a fact, they are extremely well balanced. However, some participants seem to take advantage from using one tool or the other, which at least supports the idea that the two should be available.

One step to consider in the future consists in identifying the profile of the participants that best use each tool: novices, moderately experienced, etc. Another clear step to consider in a future evaluation consists in fine tuning the model and its implementation to avoid generating agendas with too many tasks.

6   Conclusions and Future Work

This paper describes the Facilitation Tool, a tool that assists facilitators managing GDSS. Two design issues were of particular consideration. The first one is that facilitators must carefully plan decision-making processes in advance, a task that requires either past experience or some degree of assistance. The second subject concerns remote facilitation, a problematic situation limiting facilitators' interventions caused by the low level of media richness.

The Facilitation Tool is based on a comprehensive model of the decision-making process, which guides the whole planning activities starting from a high-level perspective down to the selection and configuration of specific GDSS tools that best fit the problem at hand.

Concerning remote facilitation, the Facilitation Tool implements a set of techniques covering three types of interventions in the decision-making process: steer and focus group participants, analyse and understand issues, and moderate conflicting or chaotic situations.

Currently, we have tested the planning functionality with a set of four facilitators and two decision problems. A comparison with GroupSystems' agenda has been made but is not yet conclusive. Subjects produced meeting agendas that differ significantly from agendas generated with GroupSystems. In particular, the Facilitation Tool seems to generate a greater number of tasks for the same problem. On the one hand, FT reduces agenda structure and efficiency but, on the other hand, increases logical sequence and clarity.

Regarding future work, our intention is to further develop the pre-meeting assistance provided by the Facilitation Tool, integrating a set of decision cases with the decision-making model, serving as examples of best practice. Also, our intention is to open the decision-making model to the participants, as a way of improving future group performance.

Remote facilitation has not yet been subject of controlled experiments, although a comparison with audio/videoconferencing is being prepared.

7   Acknowledgements

8   Bibliography

[1].   Aiken, M., Motiwalla, L., Sheng, O., Nunamaker, J., 1990. ESP: An expert system for pre-session group decision support systems planning. Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences. January.

[2].   Aiken, M., Sheng, O., Vogel, D., 1991. Integrating expert systems with group decision support systems. ACM Transactions on Information Systems, v. 9 n. 1. January.

[3].   Antunes, P., Guimaraes, N., Segovia, J., Cardenosa, J., 1995. Beyond formal processes: Augmenting workflow with group interaction techniques. Conference on Organizational Computing Systems (COOCS '95). ACM Press. San Jose, California. August.

[4].   Clawson, V., Bostrom, R., Anson, R., 1993. The role of the facilitator in computer-supported meetings. Small Group Research. November.

[5]. Costa, C., Ho, T., Antunes, P., 1999. Facilitation Organisational Activities Using Plans and Audits. First International Conference on Enterprise Information Systems. Setubal, Portugal. March.

[6]. Dennis, A., George, J., Jessup, L., Nunamaker, J., Vogel, D., 1988. Information technology to support electronic meetings. Management Information Systems Quarterly. December.

[7]. Dickson, G., Poole, S., DeSanctis, G., 1992. An overview of the GDSS research project and the SAMM system. Bostrom, Watson, Kinney (editors), Computer Augmented Teamwork: A guided tour. Van Nostrand Reinhold.

[8]. Dickson, G., Partridge, J., Robinson, L., 1993. Exploring modes of facilitative support for GDSS technology. Management Information Systems Quarterly, June.

[9]. Dubs, S., Hayne, S., 1992. Distributed facilitation: A concept whose time has come? Proceedings of ACM CSCW '92 Conference on Computer-Supported Cooperative Work, Toronto, Canada, November.

[10]. Gallupe, R., DeSanctis, G., Dickson, G., 1988. Computer-based support for group problem finding: An experimental investigation. Management Information Systems Quarterly. June.

[11]. Hwang, C. and Lin, M., 1987. Group decision making under multiple criteria: Methods and applications. Springer-Verlag. 1987.

[12]. Jarvenpaa, S., Rao, V., Huber, G., 1988. Computer support for meetings of groups working on unstructured problems. MIS Quarterly. December.

[13]. Johansen, R., Sibbet, D., Benson, S., Martin, A., Mittman, R., Saffo, P., 1991. Leading business teams: How teams can use technology and group process tools to enhance performance. Addison-Wesley, Organization Development Series.

[14]. Kaner, S., 1996. Facilitator's guide to participatory decision-making. New Society Publishers.

[15]. Lewicki, R., Weiss, S. and Lewin, D., 1992. Models of conflict, negotiation and third party interventions: A review and synthesis. Journal of Organizational Behavior, V. 13.

[16]. McCartt, A., Rohrbaugh, J., 1989. Evaluating group decision support system effectiveness: A performance study on decision conferencing. Decision Support Systems. June.

[17]. McGrath, J.E., 1984. Groups: Interaction and performance. Englewood Cliffs, NJ, Prentice Hall.

[18]. Ngwenyama, O., Bryson, N., Mobolurin, A., 1996. Supporting facilitation in group support systems: techniques for analyzing consensus relevant data. Decision Support Systems, Vol. 16.

[19]. Niederman, F., Beise, C., Beranek, P., 1996. Issues and concerns about computer-supported meetings: The facilitator's perspective. Management Information Systems Quarterly. March.

[20]. Niederman, F., Volkema, R., 1996. Influence of agenda creation and use on meeting activities and outcomes: Report on initial results. Proceedings of the 1996 conference on ACM SIGCPR/SIGMIS conference. Denver Colorado.

[21]. Nunamaker, J., Dennis, A., Valacich, J., Vogel, D., George, J., 1991. Electronic meeting systems to support group work: theory and practice at Arizona. Commmunications of the ACM, Vol. 34, 7.

[22]. Nunamaker, J., Dennis, A., George, J., Martz Jr., W., Valacich, J., Vogel, D., 1991. GroupSystems. Bostrom, Watson, Kinney (editors), Computer Augmented Teamwork: A guided tour. Van Nostrand Reinhold.

[23]. Nunamaker, J., R. Briggs, D. Mittleman, D. Vogel and P. Balthazard, 1997. Lessons from a dozen years of group support systems research: A discussion of lab and field findings. Journal of Management Information Systems, 13(3).

[24]. Schwarz, R., 1994. The skilled facilitator. Jossey-Bass Publishers.

[25]. Simon, H., 1997. Administrative behavior: a study of decision-making processes in administrative organizations (4th edition). Simon & Schuster Inc.

[26]. The 3M Meeting Management Team, 1994. Mastering meetings. McGraw-Hill, Inc.

[27]. Viller, S., 1991. The group facilitator: A CSCW perspective. Proceedings of the Second European Conference on Computer-Supported Cooperative Work. Amsterdam, The Netherlands.

[28]. Zigurs, I., Poole, M., DeSanctis, G., 1988. A study of influence in computer-mediated group decision making. Management Information Systems Quarterly. December.

[29]. www.reactivesystems.com

[30]. www.covision.com

[31]. www.facili.com