

RADBOD UNIVERSITY



MASTER THESIS

Distributed Group Model Building

A Dialogue Games approach using Slack

Author:
M.J. (Mats) OUBORG

Supervisor:
dr S.J.B.A. (Stijn)
HOPPENBROUWERS

Second examiner:
prof.dr. E.A.J.A. (Etiënne)
ROUWETTE

*A thesis submitted in fulfillment of the requirements
for the degree of Master of Science in the Information Sciences*

at the

Institute for Computing and Information Sciences

October 2017

“Wie de vorm beheerst, is de inhoud meester.”

— Frederik (Frits) Bolkestein

RADBOUD UNIVERSITY

Abstract

Faculty of Science
Institute for Computing and Information Sciences

Master of Science in the Information Sciences

Distributed Group Model Building

A Dialogue Games approach using Slack

by M.J. (Mats) OUBORG

Globally operating organizations sometimes encounter Messy Problems (Ackoff, 1974, 1979), which are problems that consist of complex interacting systems that are not easy to explain. Therefore finding a solution for these Messy Problems can be difficult. Group Model Building (Richardson and Andersen, 1995) (Vennix, 1996) is a method to tackle these Messy Problems. Group Model Building invites relevant stakeholders in the same room together and helps them to collaboratively create a causal loop diagram, a stock&flow diagram, simulations, or all three, which creates an overview and therefore an understanding of the Messy Problem they encounter. Group Model Building uses different methodologies to prevent bias and politics from influencing this result, as that might create a distorted overview of the Messy Problem. The necessity for all the relevant stakeholders to be in the same room together can be seen as a disadvantage for globally operating organizations, because stakeholders initially might be far away from each other. To save traveling time and therefore improve reaction time when solving Messy Problems, this thesis explains the creation of a distributed approach of Group Model Building. This Distributed Group Model Building approach is build on the Slack platform which enables participants to use Group Model Building without the necessity of being in the same room together. Additionally, the distributed approach can also be used as an administrative aid for regular Group Model Building sessions, which makes it more easy to register choices that have been made. It also creates a digital overview of the information that has been created.

Acknowledgements

I would like to thank a couple of people for their helpfulness during the creation of this thesis:

- Stijn Hoppenbrouwers & Etiënne Rouwette for their support, advice, and supervision.
- Bart Gruppen, Willem Boumans, Luuk Scholten, Ruud Seegers, Robert Hissink Muller, Monic Lansu, Pleun van Arensbergen, and Linda Nijland for their testing efforts of the tool created in this thesis.
- Gerdriaan Mulder, Thom Wiggers, Luuk Scholten, and Wietse Kuipers for their technical support during the development on the Slack-platform.
- Thom Wiggers for his linguistic support during the writing of this thesis.

And all others who helped me during the ups and downs encountered during the development of this research. Without all of you, this project would never have been finished with the success it has now.

Thank you!

Contents

Abstract	iii
Acknowledgements	iv
Contents	v
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 About this thesis	1
1.2 Research question	2
1.3 Content of this thesis	5
2 Literature	7
2.1 Solving Messy problems	7
2.1.1 Messy Problems	8
2.1.2 Group Model Building Methodology	10
2.2 Group Model Building as a formalized process	15
2.2.1 The process	16
2.2.2 Criticism	19
2.3 Basic requirements for Group Model Building	20
2.4 The Dialogue Games approach	27
2.5 The combined approach for Distributed Group Model Building	30
2.5.1 Scaffolding & ThinkLets	30
2.5.2 Combined approach	31
3 Using Slack	35
3.1 About Slack	35
3.1.1 Basic functions	36
3.1.2 A Slack Bot	37
3.2 Translating the literature into a Slack App	37
3.2.1 Slack Bot role within Group Model Building	38
3.2.2 Facilitators role within Slack	38
4 The Sequential Group Model Building Bot	41

4.1	Technical choices	41
4.2	The sequential Distributed Group Model Building tool	42
4.2.1	Open and define rules	43
4.2.2	State the observed problem	43
4.2.3	Generate a detailed view of the problem variable	44
4.2.4	Create causal relationships	45
4.2.5	Calculate and define end result	45
5	First impression in practice	47
5.1	Testing the sequential implementation	47
5.2	General reception	48
6	The final implementation	51
6.1	Implementing the combined approach	51
6.1.1	Open and define rules	52
6.1.2	State the observed problem	53
6.1.3	Generate a detailed view of the problem	58
6.1.4	Create causal relationships	62
6.1.5	Calculate and define end result	65
6.2	Considerations	66
7	Testing the final implementation	69
7.1	Designing the last tests	69
7.2	Results after testing	70
8	Results and discussion	75
8.1	Completing the design cycle	75
8.2	Is the tool future proof?	77
8.3	Evaluation conclusion	78
9	Further research	79
9.1	What has been produced?	79
9.2	Integrating model generation	80
9.3	Central flexibility	81
9.4	An AI facilitator	82
10	Conclusion	83
	Bibliography	85

List of Figures

1.1	Design Science Research Cycles (Hevner, 2007)	3
1.2	The regulative cycle of Design Science (Wieringa, 2009)	4
2.1	The basic process of the Nominal Group Technique (Delbecq, Van de Ven, and Gustafson, 1975)	11
2.2	Example causal loop diagram (system dynamics) on early release policy (Rouwette, Vennix, Hooff, and Jongebreur, 2007)	13
2.3	Overview of the basic Group Model Building process (Ouborg, 2015) . . .	16
2.4	Overview of the Generate a detailed view of the problem variable process (Ouborg, 2015)	17
2.5	Overview of the Create causal relationships process (Ouborg, 2015)	17
2.6	Overview of the Calculate and define end result process (Ouborg, 2015) .	18
2.7	Openers for the InterLoc implementation of Group Model Building (Hoppenbrouwers and Rouwette, 2012)	29
2.8	The main process of the combined approach of Group Model Building . .	32
3.1	Basic principle of a Slack Bot	37
4.1	The basic architecture of the bot and Slack	42
6.1	The initial state of the Slack application	52
6.2	Starting of the Group Model Building bot	52
6.3	Group Model Building welcome message	53
6.4	Group Model Building session has started	53
6.5	New message from the Group Model Building bot	53
6.6	First bilateral message from the Group Model Building bot	54
6.7	Suggesting profit as the main variable	54
6.8	Main variable received and change instructions	54
6.9	Main variable message in main channel	55
6.10	Remark of participant in bilateral conversation with the bot	55
6.11	A reply to the original message	56
6.12	A thread has been opened by Slack	56
6.13	Leading user reacts to remark	57
6.14	Leading user lets bot know it can continue	57
6.15	The bot wants to start a Nominal Group Technique session	58
6.16	Nominal Group Technique started in separate threads	58
6.17	Bilateral conversation for creating detail	58
6.18	Created detail in the bilateral conversation	58
6.19	Letting the bot know a participant is ready	59

6.20	Making sure the participant is ready	59
6.21	Waiting for other participants to finish	59
6.22	Nominal Group Technique session is finished	59
6.23	Results of the Nominal Group Technique	60
6.24	Participant agrees with variable	61
6.25	Deleting a suggested variable	61
6.26	A variable has been deleted	62
6.27	Nominal Group Technique phase has ended	62
6.28	Start of the causal relations phase	63
6.29	Control panel for the causal relations phase	63
6.30	Variable of interest is chosen	63
6.31	Thread for cause suggestion	64
6.32	Overview after a discussion	65
6.33	Overview after a the session has finished	65
8.1	The regulative cycle of Design Science (Wieringa, 2009)	75
9.1	Example causal loop diagram (system dynamics) on early release policy (Rouwette et al., 2007)	80

List of Tables

2.1	Requirement 0: Basic Group Model Building steps.	21
2.2	Requirement 1: Basic Open and define rules actions.	21
2.3	Requirement 2: Basic State the observed problem actions.	22
2.4	Requirement 3: Basic Generate a detailed view of the problem variable steps.	22
2.5	Requirement 4: Basic Create causal relationships steps.	22
2.6	Requirement 5: Basic Calculate and define end result steps.	23
2.7	Requirement 3a: Preparation of Generate a detailed view of the problem variable steps.	23
2.8	Requirement 3b: Underpinning of shared variables of Generate a detailed view of the problem variable steps.	24
2.9	Requirement 4a: Use the detail to find causes and effects of the problem variable of the Create causal relationships steps.	24
2.10	Requirement 4b: Check for feedback loops of the Create causal relationships steps.	25
2.11	Requirement 6: Facilitation requirements.	26
2.12	Requirement 6a: Facilitation requirements.	26

Chapter 1

Introduction

1.1 About this thesis

This thesis is about creating a distributed tool that supports group decision making, namely *Distributed Group Model Building*, and is submitted in fulfillment of the requirements for the degree of Master of Science in the Information Sciences at the Radboud University Nijmegen. Regular Group Model Building is a team collaboration method used to tackle Messy Problems (Ackoff, 1974, 1979), which are defined as problems when *“managers are not confronted with separate problems but with situations that consist of complex systems of strongly interacting problems”* (Ackoff, 1974, 1979). With the creation of a causal loop diagram as its main goal (Vennix, 1996) (Rouwette and Franco, 2014), Group Model Building as a method tries to tackle these Messy Problems by creating an overview of the different complex interacting systems.

Group Model Building can be used as a team collaboration activity, by inviting all the relevant stakeholders into a session in a single room and letting these stakeholders jointly build this causal loop diagram (Vennix, 1996) (Rouwette and Franco, 2014). In contrast to regular Group Model Building (Richardson and Andersen, 1995) (Vennix, 1996) (Rouwette and Franco, 2014), Distributed Group Model Building will enable participants to enact in sessions without the necessity to be in the same room together. This creates new possibilities for Group Model Building, since it then can be used online, but also in a distant manner.

When the distributed tool of this thesis is used online, it can be used as an additional administrative aid during the regular Group Model Building process (for instance to keep track of decisions that have been made). In other words, the tool can be used as an addition to a normal (single room) approach of Group Model Building. On the other hand, participants can chose not to be in the same room together and use the tool as its main communication system, which means all the possible communication of a Group Model Building session will go through the tool. Given that a regular Group Model Building session is always held in one location, the benefits of the tool makes it possible for Group Model Building to be used in a distributed fashion. Since many companies operate globally these days with different stakeholders at a far distance from each other (Robbins and Barnwell, 2006), the distributed nature of the tool will make it possible for these companies to use Group Model Building as well when they tackle a Messy Problem.

The notion of the need for a Distributed Group Model Building method, was first described by Ouborg (2015). In that study, a formalized approach was chosen to get hold of the processes and data of Group Model Building. These processes and data would in return act as a blueprint for a digital approach, which in return could be the basis for a distributed approach to Group Model Building. The study resulted in a large set (about 172) of models which described a sequential process of Group Model Building. In his conclusion however, Ouborg (2015) criticized that this sequential approach resulted in a way too complex blueprint of Group Model Building, which made it infeasible to realize. One can also state that the sequential approach of Group Model Building might not fully represent its collaborative nature. Group Model Building was not thought of as a fully sequential process (Vennix, 1996), and it is up to the facilitator of the sessions to choose how much formality is thought necessary (Vennix, 1996) (Rouwette and Franco, 2014).

As an inspiration for a more agile approach, Hoppenbrouwers and Rouwette (2012) is chosen as a new direction. Their approach of Group Model Building is that of Dialogue Games, which is an approach in which only the boundaries of a process are defined (instead of defining the correct path that must be followed). This enables more freedom for the participants when using Group Model Building, without doing steps that are not allowed in Group Model Building. Hoppenbrouwers and Rouwette (2012) created a pioneering distributed form of Group Model Building using the dialogue games tool InterLoc (Ravenscroft and McAlister, 2006). This thesis tries to build upon the works of Ouborg (2015) and Hoppenbrouwers and Rouwette (2012) by combining the detail that was created (Ouborg, 2015), with the dialogue games and the less sequential approach (Hoppenbrouwers and Rouwette, 2012) by creating a chat bot in Slack. This is done by examining the context of Group Model Building and facilitation first, which is then translated into the practical procedures of Slack.

1.2 Research question

Given that this research is conducted as a master thesis, it requires extra focus on a research approach. The research of Ouborg (2015) lacked the explanation of the research methodology used. Although this does not mean that the quality of research in the work of Ouborg (2015) is low, a formal check of its approach is not possible. What was done in his research was simply a translation from literature into formal models, which in retrospect can be defined as a methodology. If we look into the most respected methodology that mostly represents the methodology of Ouborg (2015), we find the Design Science methodology (Hevner, March, Park, and Ram, 2004). Design Science was also the methodology used by Hoppenbrouwers and Rouwette (2012). Given the lack of explanation of the research methodology used in Ouborg (2015), we now clearly state that Design Science will be the methodology for this research. In order for Design Sciences to be used soundly, it is important to explain what it is and how we use it in this thesis research.

Design Science is a research methodology that has its origins in research methodologies that are used in Information Systems (Hevner et al., 2004, Hevner, 2007) (Iivari, 2007) (Wieringa, 2009). Hevner et al. (2004) created a framework for this methodology and described it later as the three cycles of Design Science (Hevner, 2007). These cycles explain the Design Science Research and its two main contexts based on *Environment*

and *Knowledge Base* (Hevner, 2007). The Environment represents the stakeholders, the organizational and technical systems, and the problems and opportunities that are involved, which combined are called the *Application Domain* (Hevner, 2007). The Knowledge Base represents the Scientific Theories and Methods, the Experience and Expertise, and the Design Products and the Design Processes that are involved, which combined are called the *Foundations* (Hevner, 2007).

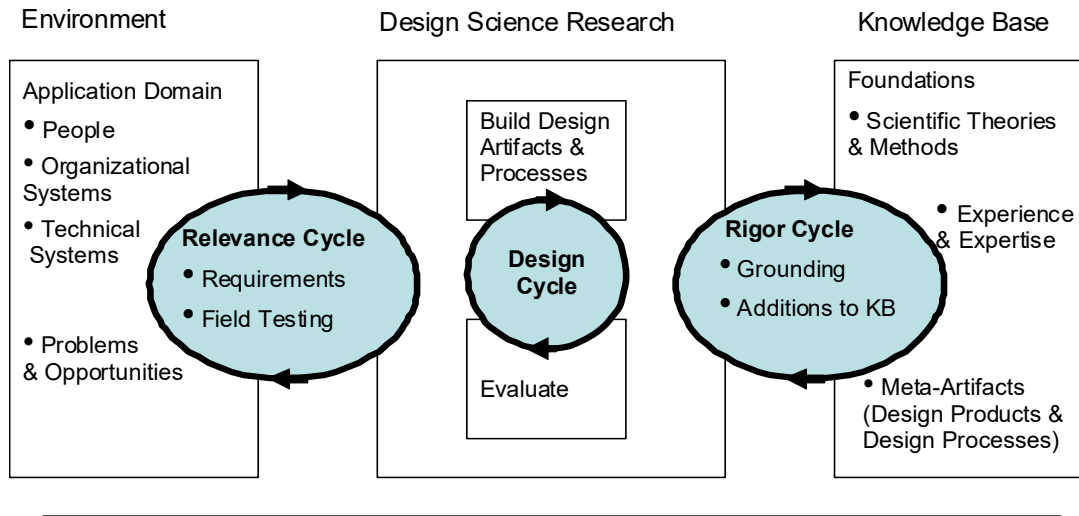


FIGURE 1.1: Design Science Research Cycles (Hevner, 2007).

Besides the Design Science Research, and its two main context fields, there are three cycles that connect these three worlds. First, there is an internal cycle called the *Design Cycle* which explains the main steps in Design Science, namely Building and Evaluating. The Design Science Research field is connected to the Environment field via the *Relevance Cycle* which uses the constant creation and testing of Requirements. Last, the *Rigor Cycle* connects the Knowledge Base to the Design Science Research by constantly grounding the current research on earlier and related research and by adding new research results to the Knowledge Base. This whole process is viewed in Figure 1.1 (Hevner, 2007).

Later, Wieringa (2009) extended the Three Cycle Model by creating a framework that "clarifies the interface of design science with its social environment and with the scientific knowledge base" (Wieringa, 2009). It does so by creating guidelines for a design science researcher, which are as follows (Wieringa, 2009):

1. *Distinguish practical problems from knowledge questions* In practical problems stakeholders desire to change the world, in knowledge questions the researcher desires to change his/her knowledge of the world.
2. *Solve practical problems by the regulative cycle* As shown in Figure 1.2 (Wieringa, 2009).
3. *Distinguish problem investigation from design validation* In problem investigation existing phenomena are investigated, in design validation the effects of an unimplemented design are predicted.

4. *Problem investigation may be problem-driven, solution-driven, goal-driven, or impact-driven* In problem investigation, one or more of these tasks needs to be done: diagnose a problem, operationalize goals, check the validity of the design argument, or investigate the impact of realized implementations.
5. *When designing a solution, maintain the design argument* The causation part of the design argument says that the solution in a context will have certain effects, the valuation part says that these effects satisfy stakeholder criteria.
6. *When validating a design, consider trade-offs and sensitivity* In trade-offs we vary the solution, in sensitivity analysis we vary the environment.
7. *When validating a design, aim to incorporate conditions of practice* Scale up from controlled conditions to realistic conditions.
8. *When solving a knowledge question in the regulative cycle by means of research, no research method is banned* Research design must be justified, as anywhere else, in terms of research questions, the investigated domain and available resources to do the research.

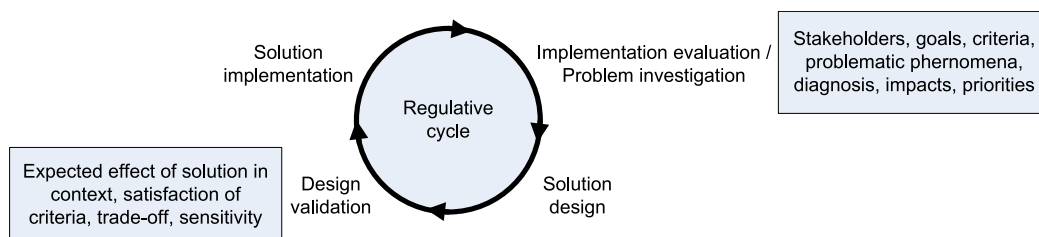


FIGURE 1.2: The regulative cycle of Design Science (Wieringa, 2009).

The Three Cycles (Hevner, 2007) and the Design Science Guidelines (Wieringa, 2009) creates the basis for the research methodology used in this thesis. The Cycles will be used to define two basic research questions. The Guidelines will be used on the fly throughout this thesis, and during the presentation of the results to explain what the results are. The two research questions are in the form of a Design and an Evaluation question and are as follows:

- **Design Question:** *What are the basic requirements for Group Model Building?*

Before the Slack app can be build, we have to know what at least is required (in terms of process and data) in order to contain the Group Model Building benefits. This question can also be interpreted as the need for a list of requirements of Group Model Building (i.e. what is necessary to be able to call it Group Model Building). Previous literature should form the basis in answering this question, therefore, it is answered mainly in Chapter 2.

- **Evaluation Question:** *Does a Distributed Group Model Building tool using Slack enable the benefits of regular Group Model Building?*

After the Slack app is built, it should of course be tested in practice. This is to indicate if the distributed approach using a chat tool like Slack creates at least the same benefits of regular Group Model Building. Also an approach using Slack

might emerge new benefits for Group Model Building, or clearly state potential pitfalls. The indication will be made in terms of questionnaires and video footage of live usage which will be briefly reviewed and interpreted. This research question can therefore only be mainly answered at the end of this thesis, which is done in Chapter 8 and Chapter 10.

Given this thesis, its research questions and its inspiration of Ouborg (2015), it is important to explain its link to the Design Science methodology more concretely. The initial need for a distributed form of Group Model Building can be explained as Problem & Opportunity from the Environment within the Design Science methodology. In Ouborg (2015) this was done by setting up one basic requirement, namely *Formalize the processes and data of Group Model Building*. Within the Relevance Cycle this is the basic requirements step. This continued within the Design Science, by following the Building phase within the Design Cycle. But that could only be done by finding enough literature that described the Group Model Building processes in greater detail. Therefore, the Rigor Cycle was immediately used to ground the theories necessary for a Formal Group Model Building approach. The combined result of this process was therefore a list of BPMN and ORM models, which in return would be the addition to the Knowledge Base, which completes the Rigor Cycle.

Although this was achieved, Ouborg (2015) lacked in completing the Design Cycle and Relevance Cycle by not practically evaluating the results that were generated. To evaluate if the models generated by Ouborg (2015) are sound enough, the first Slack App will be just a copy of the processes created in that research. It is then tested in practice and depending on the results, the Design Cycle would continue by returning to the Build phase which in return would be evaluated. Given that the basic critique on Ouborg (2015) is that it follows a sequential path instead of a more agile approach, it is expected to follow the Design Cycle at least one more time. This is all explained in the later chapters of this thesis.

1.3 Content of this thesis

To structure the research in this paper, it is built as follows. First, a general overview of Group Model Building, Dialogue Games, and their context is given (Chapter 2). Then, Slack is introduced as a possible framework to build the Distributed Group Model Building tool (Chapter 3). Next, a basic implementation of a Group Model Building Slack bot is being created and tested (Chapter 4 and Chapter 5). Furthermore an improved version is built and tested (Chapter 6 and Chapter 7). Last, results are discussed (Chapter 8), further research suggestions are given (Chapter 9) and conclusions are given (Chapter 10).

Chapter 2

Literature

This chapter contains the literature research that was conducted prior to the creation of the *Group Model Building Slack Bot*. By examining the origins, context and specifics of Group Model Building in general, a basis is created that describes the requirements for Group Model Building and the potential pitfalls when used wrongly.

Ouborg (2015) already gave an overview of Group Model Building in general, but the formal processes that Ouborg (2015) derived are limited to a simple translation of the Group Model Building process into modeling languages. This literature research however, is more focused in defining a broader context of Group Model Building.

The broader context of Group Model Building is generated by not only looking into its foundations, but also by looking into related facilitation in general. These all form the basis for a list of requirements of Group Model Building (Section 2.5), which in return forms the basis for the Slack Bot (Chapter 3).

This chapter is build as follows: First, *Messy problems* are explained, and additional Group Model Building methodology is given (Section 2.1). Then, the formal processes created by Ouborg (2015) are elaborated (Section 2.2.2). Next, a list of requirements for Group Model Building is given (Section 2.3). Furthermore, the research into dialogue games by Hoppenbrouwers and Rouwette (2012) is elaborated (Section 2.4). Last, two more principles (Scaffolding & ThinkLets) are being elaborated and all the theory is combined into a new approach (Section 2.5).

2.1 Solving Messy problems

The goal of Group Model Building can be explained in two different ways, namely from a *problem solving perspective* and an *information generation perspective*. The problem solving aspect is about solving *messy problems* (Ackoff, 1974, 1979), with which complex and difficult to define problems are meant (see Section 2.1.1). Solving messy problems is for many users the main reason to use Group Model Building, as it has proven to be a great tool to do so (Vennix, 1996) (Rouwette, Vennix, and Mullekom, 2002) (Rouwette et al., 2007) (Rouwette and Franco, 2014).

The information generation perspective talks about the data that is produced by a Group Model Building session. Hoppenbrouwers and Rouwette (2012) mention that

Group Model Building is used to find causal relations in a gradual way. This is due to the fact that Group Model Building is always based on System Dynamics (Forrester, 1961, 1975) (Vennix, 1996) (Rouwette and Franco, 2014). It does so by letting the group build on a causal loop diagram (Vennix, 1996) (Rouwette et al., 2002) (Rouwette et al., 2007) (Rouwette and Franco, 2014).

Both the information generation and the problem solving aspect are important in order to be able to fully understand Group Model Building in general, therefore both will be explained. Given that Solving Messy Problems is a goal before one uses Group Model Building, we start by explaining what Messy Problems are exactly. Then, the basic methodology that Group Model Building is based on (namely, System Dynamics) is explained. Also, other methodologies that Group Model Building uses are explained.

2.1.1 Messy Problems

A problem is called a *messy problem* when *"managers are not confronted with separate problems but with situations that consist of complex systems of strongly interacting problems"* (Ackoff, 1974, 1979). Ouborg (2015) gives an example in which a crisis within a large company suddenly occurs. This potential panic situation calls for a general meeting in which all key figures of the company are present. Here they debate the problem and they try to find a solution. To clarify a messy problem, the example of Ouborg (2015) is cited next:

"Imagine, you are working as head of the marketing department in a multinational company, and due to the multinational nature of this company, every department is established in a different country. Your marketing department is settled in The Netherlands, the company's headquarters is settled in the US, and the company's financial office is settled in the UK. Due to declining profits, the head of the company asked every head of every department to think about a way to increase profits to save the company from going down. The CEO organizes a meeting at the headquarters in New York, where all the department bosses can establish a new strategy to prevent the company from going down.

As head of the marketing department, you prepare yourself by asking within your department if anybody can name a cause for the declining profits. As marketers, you immediately think that the declining profits is due to the lack of interesting products your company sells. Accompanied with some great new product ideas you fly to New York. You arrive just in time for your meeting. You are accompanied by the CEO, the head of the financial department, and the head of human resources. The meeting starts and the CEO asks who has come up with some ideas. You start by saying that a new product, which enthuces potential buyers, could create some new profits, which in its case will improve overall company performance.

Not long after that you find yourself in a big debate with the head of the financial department who find your idea insufficient, because a new product means extra development costs, thus lower profits. The head of the financial department thinks a great way of lowering costs is to cut on salaries, which offended the head of the human resource department. In his turn he thinks that the attraction of some clever new minds would create some fresh air, which makes you angry because of the fact that your marketing department has some of the brightest minds of the company.

Quickly the meeting transforms in to one big fight, in which is debated what should be the best solution for this problem. Mutual respect is hard to find and the CEO decides to intervene. He says that he thinks these are all great ideas to create new profit, but he does not know if we anticipate the problem well enough. The discussion goes on and on and people make a lot of fuss about each other's findings about what the problem exactly is. In the end, the discussion is cut off by the CEO to prevent that more damage is being done due to the extreme fighting discussions.

Meanwhile some months and meetings later, the profits of the company are still declining. You are still fighting with the other department heads about what causes this decline. You all think you should act fast, because the company's cash reserves are getting smaller and smaller. Still, as marketer, you do not want to get overruled by some new bright mind which would be hired by the human resource department. The financial department is not attending meetings anymore, because they think nobody listens to them. Eventually everybody thinks their idea vanishes, and your company is in a governmental crisis.” (Ouborg, 2015)

This example represent a general Messy Problem. The most significant finding is that a solution always has an indirect effect on something else. This might cause lack of control when actions are performed (Ackoff, 1974, 1979). A messy problem can further be described as follows (Rouwette and Franco, 2014) (Ouborg, 2015):

- Messy problems will have interconnectedness between different aspects of the situations (its systemic nature).
- Messy problems will exhibit high levels of uncertainty.
- Local solutions to a particular problem only generate other or new problems.

Messy Problems are connected to *Wicked Problems* (Rittel and Webber, 1973), which in some way represent a Messy Problem in greater detail (Ouborg, 2015). Wicked Problems are explained by ten characteristics (Rittel and Webber, 1973):

1. There is no definitive formulation of a wicked problem.
2. Wicked problems have no stopping rule.
3. Solutions to wicked problems are not true-or-false, but good or bad.
4. There is no immediate and no ultimate test of a solution to a wicked problem.
5. Every solution to a wicked problem is a "one-shot operation"; because there is no opportunity to learn by trial and error, every attempt counts significantly.
6. Wicked problems do not have an enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-described set of permissible operations that may be incorporated into the plan.
7. Every wicked problem is essentially unique.
8. Every wicked problem can be considered to be a symptom of another problem.

9. The existence of a discrepancy representing a wicked problem can be explained in numerous ways. The choice of explanation determines the nature of the problem's resolution.
10. The social planner has no right to be wrong (i.e., planners are liable for the consequences of the actions they generate).

These two lists provide a basic understanding of Messy Problems and helps in being able to identify them. Before a Group Model Building session is held, it is important to check if a problem is a messy one. This chapter will not further explain to what extend Messy Problems are being solved by Group Model Building, or what the social background of Messy Problems is. It is simply seen as a given in this thesis. Greater detail in the explanation of Messy Problems can be found in Ouborg (2015) and Ackoff (1974, 1979). Next section talks about the methodologies used by Group Model Building that helps solving Messy Problems.

2.1.2 Group Model Building Methodology

In order to be able to solve Messy Problems, Group Model Building uses two central subjects, namely:

- System Dynamics (Forrester, 1961, 1975), which is the foundation of Group Model Building
- Facilitation principles (Schein, 1990, 1997) (Vennix, 1996) (Rouwette and Franco, 2014), which are used during the execution of Group Model Building

Furthermore in order to prevent bias during the creation of ideas, Group Model Building uses:

- Nominal Group Technique (Delbecq et al., 1975)

These subjects all have their own benefits in helping solving Messy Problems. Nominal Group Technique helps in overcoming political or individual bias in the end result, System Dynamics is used to describe the system of interacting systems of the Messy Problem (or the current situation of the organization), and Facilitation Principles create a sharing atmosphere and in the end helps building consensus. Because all three are important, they are all explained below.

It should be noted that in general System Dynamics and Facilitation form the foundation of a Group Model Building session. The Nominal Group Technique is an important aspect of the idea creation phase, but can be explained as a technique used by Group Model Building instead of it being a foundation of Group Model Building. Because the Nominal Group Technique in our opinion is highly suitable for digitization, we will explain it here as well.

Nominal Group Technique

The *Nominal Group Technique* (Delbecq et al., 1975) is presented as an alternative to brainstorming (Osborn, 1953, 1957, 1963). Brainstorming has a couple of disadvantages, namely: free riding, social loafing, social inhibition, and production blocking (Stroebe, Nijstad, and Rietzschel, 2010). These disadvantages, which enable a Messy Problem to become even bigger, can be overcome when using Nominal Group Technique. It does so by splitting idea creation from idea sharing (Delbecq et al., 1975) (Rouwette and Franco, 2014) (Ouborg, 2015).

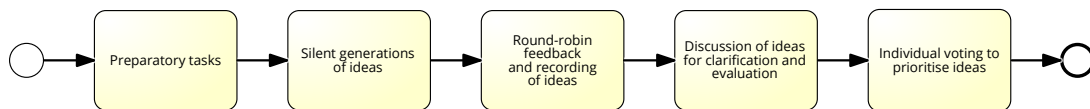


FIGURE 2.1: The basic process of the Nominal Group Technique (Delbecq et al., 1975).

Figure 2.1 shows the main steps of the Nominal Group Technique in BPMN¹. Only when these steps are followed one-on-one, the disadvantages of brainstorming can be overcome. Good Nominal Group Technique is controlled by a facilitator (or mediator) who is aware of these steps and the session itself is conducted while participants are sitting in a U-shape in front of a whiteboard (Delbecq et al., 1975) (Ouborg, 2015). To explain what is done for each step, the explanation given in Ouborg (2015) is shown below (Delbecq et al., 1975):

- *Preparatory tasks*: In this step the mediator makes sure that the meeting room is set in the previously mentioned U-shape setting. He or she also checks if there is a piece of paper and a pen for every participant. Then he or she welcomes the participants and thanks them for participating in this idea generation meeting. The mediator writes down the problem variable for which some possible solutions must be generated. He or she asks the participants if they understand the problem variable.
- *Silent generation of ideas*: Here, the mediator explains that participants now will write down some ideas they are having. He or she makes it absolutely clear that it is forbidden to talk to each other for five minutes, while they write down some ideas individually. The mediator also states that every idea is welcome and should be written down on the paper in front of the participants in a brief manner. Catchwords or tiny phrases are preferred. The mediator asks if everybody is ready and then starts the five minutes of silent idea generation.
- *Round-robin feedback and recording of ideas*: After the five minutes the mediator asks the participants to lay down their pens. Talking to each other is still not allowed. The mediator now tells the participants that he or she will ask every participant to read one idea he has come up with, and when shared, the mediator will write down the idea on the whiteboard. This will happen in a round robin fashion until all the ideas have been shared.
- *Discussion of ideas for clarification and evaluation*: After all the ideas have been shared and written down on the whiteboard, the mediator will tell the participants they can ask questions if they want clarification or explanation about the ideas that are written down. The mediator then ask the group to clarify.

¹Business Process Modeling Notation (Weske, 2010).

- *Individual voting to prioritize ideas*: When everything is clear the participants are asked to write down on their piece of paper a top 10 of most important ideas. This is used as a voting system to create a top many of the best ideas to prioritize them. Other prioritization methods are possible.² When this is all done, the mediator promises he will make a report and sends it to the participants. He then thanks them and closes the session.

This concludes the explanation of the Nominal Group Technique. For a broader background of the nominal group technique we refer to Ouborg (2015). In Section 2.2.2 is explained how this technique is implemented in Group Model Building. This also explains how a digitized version of the Nominal Group Technique might look like.

System Dynamics

System Dynamics (Forrester, 1975) (formerly called *industrial dynamics* (Forrester, 1961)) is a modeling language where "the characteristics of the whole are more important than the characteristics of individual parts." (Rouwette and Franco, 2014). The most important notion of System Dynamics is that "Structure drives behavior" (Vennix, 1996) (Rouwette and Franco, 2014), therefore understanding the structure of the problem would create insight in controlling its behavior. Given that Messy Problems tend to have an escalating nature (because the source is not tackled) (Vennix, 1996) (Rouwette and Franco, 2014) (Ouborg, 2015), Group Model Building tries to find feedback loops (Vennix, 1996) (Rouwette et al., 2007) (Rouwette and Franco, 2014) (Ouborg, 2015). System Dynamics is an ideal language to find feedback loops as it connects different variables via a positive or a negative relation.³ If a loop is found, these relation types can be used to determine if there is an escalating positive or an escalating negative feedback loop (Rouwette and Franco, 2014) (Ouborg, 2015).

An example of a simple System Dynamics *causal loop diagram* would be as follows: If one has a company the most interesting variable to have might be *profit*. *Profit* might have as positive cause *revenue* and as negative cause *cost*. If more *profit* means larger investment in advertisement, then *advertisement budget* is a positive effect of *profit*. But, *advertisement budget* is also a positive cause of *revenue* and a negative cause of *cost*. This would therefore create two feedback loops: A negative feedback loop which depends on the influence of *advertisement budget* on *cost* and a positive feedback loop which depends on the influence of *advertisement budget* on *revenue*.

Although this would create a simple *causal loop diagram* which is a possible product of the system dynamics view of the organization, it would be of great information value to the organization as this would clarify that *advertisement budget* has to be researched in more detail. A better understanding of *advertisement budget* would give insight into when it has a potential more desired effect on *revenue* and *cost* (namely high *revenue* and lower *cost*).

Figure 2.2 shows a more complex causal loop diagram created by Rouwette et al. (2007). The model describes crime in The Netherlands, more specifically, how judges are effected by the situation of prisons and the history of criminals (Rouwette et al., 2007). It was

²Like clustering in two or more groups. Because voting is not part of the nominal group technique within group model building, further explanation of possible prioritization methods is left out.

³Relations can also be of an unknown type, but these are unwanted. For instance, participant might see variable A as a cause of variable B, but do not know if this is a positive or negative cause. Usually this means the relation between A and B has to be extended with extra variables to explain its relation type in more detail.

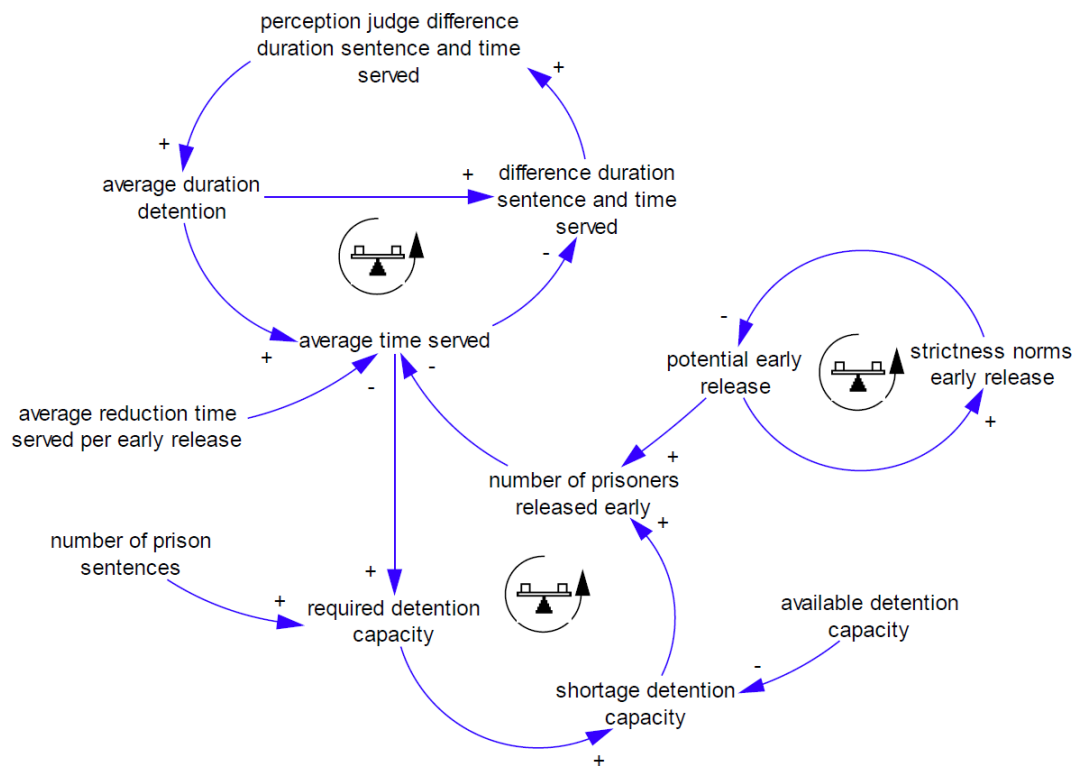


FIGURE 2.2: Example causal loop diagram (system dynamics) on early release policy (Rouwette et al., 2007).

clearly a messy problem and therefore a system dynamics approach was used to describe the situation. The arrows in the causal loop diagram describes the positive or negative relations, and the seesaw describes a feedback loop (Rouwette et al., 2007) (Ouborg, 2015). This model can therefore give an insight into how System Dynamics works.

The System Dynamics methodology is used in Group Model Building as it is a great way to describe Messy Problems. The fact that you therefore can tackle feedback loops, give the basis for the solution to tackle the Messy Problem. Therefore, general Group Model Building uses this System Dynamics approach in its method. How a causal loop diagram as product of the system dynamics approach is being build, is explained in Section 2.2.2.

Facilitation principles

An important part of Group Model Building, or even of all collaborations techniques, is how it is mediated. Mediation can be done in a couple of ways. For instance, one can choose to be leading in which he absorbs all the information that is there and decides what direction the group should go. On the other hand, a mediator can choose to be neutral in the sense that he or she only guides the process and does not choose sides. Given that Group Model Building is often used in political sensitive or interest biased situations, the leading mediator is unwanted because this will only amplifies the bias or sensitivity. Therefore Group Model Building has chosen for the latter, the more neutral unbiased mediator. In that situation that person would be called a *facilitator* (Vennix, 1996) (Rouwette and Franco, 2014).

The only role of a facilitator is to *facilitate* (as one would guess), but what should his or her attitude be? Literature describes three models of attitude (Schein, 1990), namely: providing expert information, playing doctor, and process consultation. The three differ in how much information you give to your client, or how much information you let them gather for themselves (or is relevant to them). For *providing expert information*, you are the oracle on the subject of interest, therefore the client is highly dependable on the information you give. This means that you as a facilitator more or less steer the problem solving trajectory by giving information, which can happen without anyone asking for it. This method sheds to a leading mediator, and therefore is a potential pitfall for finding an explanation for a messy problem (Schein, 1990) (Vennix, 1996) (Rouwette and Franco, 2014).

The second style, that of *playing doctor*, is somewhat less of an oracle principle. As Schein (1990) states: "*First, the doctor model assumes that the client has correctly identified the sick area. Second, it assumes that the 'patient' will reveal the information necessary for a good diagnosis. (...) Third, a correlated assumption that applies especially to consulting clinical psychologists is that they have the expertise necessary to arrive at a correct diagnosis. Fourth, this model assumes that the client will accept the diagnosis arrived at. Fifth, it assumes that the client will accept the prescription and do what the 'doctor' recommends. And finally, it assumes that the client will be able to remain healthy after the doctor leaves.*" (Schein, 1990).

This playing doctor approach therefore is only suitable for Group Model Building on the basis of process guidance. If the "doctor" is sure that it is wise to go back to the Nominal Group Technique phase instead of continue in the Group Model Building process, he or she should suggest it to the group. But, he cannot force them to do so, nor can he claim a hundred percent that this will solve the Messy Problem (which is more of an substantive statement, instead of a process one). Therefore, the third style, *process consultation* might be most interesting for Group Model Building.

To overcome bias and sensitivity within debatable discussions, you as a facilitator must be as nonjudgmental as can be (Vennix, 1996) (Rouwette and Franco, 2014) (Ouborg, 2015). Therefore, only the guidance of a process is what should be done. This is explained in the principle of *process consultation*. Schein (1990) states that this will "*better fit human systems with which we typically deal*" (Schein, 1990). In later research he explains the role of a process consultant in more detail (Schein, 1997):

- *Always be helpful*: A consultant must always do what is best. Also if this means to do less. Therefore, "*every contact should be perceived as helpful.*"
- *Always deal with reality*: "*You cannot be helpful if you do not know the realities of the client system; therefore, every contact should bring to the surface diagnostic information about the state of the client system.*"
- *Access your ignorance*: "*You cannot determine what is the current reality if you do not get in touch with what you do not know about the situation and have the wisdom and the courage to ask about it.*"
- *Everything you do is an intervention*: "*Even though the goal of exploratory inquiry through accessing ignorance is diagnostic information, the reality is that every question or inquiry is at the same time an intervention and must be treated as such.*"

- *The client owns the problem and the solution: "The reality is that only the client has to live with the consequences of the problem and the solution, so the consultant must not take the monkey off the client's back."*
- *Go with the flow, but seize targets of opportunity: "All systems develop cultures and attempt to maintain their stability through maintenance of those cultures. Therefore, one must "go with the flow". At the same time, all systems have areas of instability where motivation to change exists. One must build on existing motivations and cultural strengths, and seize targets of opportunity."*
- *Be prepared for surprises, and learn from them: "Everything that happens is a source of new data, and everything you think you know about the client system is only a hypothesis to be tested through further interventions."*
- *Share the "problem": "Neither the client nor the consultant can fully understand the reality of the situation; defining that reality is an ongoing joint effort in terms of what to do next."*

Schein (1990) concludes within these three models: *"Periodically we all find ourselves in the role of a helper. If we are to play that role effectively, we must be conscious of the choices we make about being a process consultant, an information expert, or a doctor. (...) I believe that the process consultation model is the most appropriate way to do that."* (Schein, 1990). Therefore, for Group Model Building we can conclude that it is best to start with the style of process consultant and move to one of the others when thought necessary (Schein, 1990) (Vennix, 1996) (Rouwette and Franco, 2014). This is of course highly dependent on the skills and experience of the facilitator him- or herself (Vennix, 1996) (Rouwette and Franco, 2014).

Group Model Building is founded on System Dynamics and uses Facilitation to when performing the sessions. Furthermore it uses the Nominal Group Technique to make sure politics and bias are left out when creating new ideas. Although Vennix (1996) and Rouwette and Franco (2014) give some sort of step-wise explanation of the method, it was never done in high detail. Therefore, Ouborg (2015) translated the Group Model Building literature into a formalized process. This formalized process gives insight in the Group Model Building as a tool, and will be explained next.

2.2 Group Model Building as a formalized process

After research had concluded that a formal approach to Group Model Building was necessary to create a distributed approach to the technique (Ouborg, 2015), the processes of Group Model Building were derived formally. Ouborg (2015) used BPMN⁴ to describe Group Model Building as a sequential business process. This created a list of detailed steps that could be taken in order to perform a Group Model Building session. Given that these steps were formal, preconditions and postconditions were created. These preconditions and postconditions gave insight in the data that was flowing through the processes of Group Model Building, which was formally described in ORM⁵.

⁴Business Process Model and Notation (Weske, 2010)

⁵Object Role Modeling (Paulussen and Van der Weide, 2007)

The idea behind this formalization was that this would form the blueprint for a digital form of Group Model Building (Ouborg, 2015), which would enable Group Model Building to be used in a distributed context. Distributed here means that it is not necessary anymore for the participants to all be in the same room together while using the technique. The original Group Model Building literature (Vennix, 1996) somewhat lacks a step-wise explanation of the technique in detail. In order to be able to build a software application that operates on the basics of Group Model Building, the blueprint was needed. Therefore, Ouborg (2015) extended the detailed approach of Rouwette and Franco (2014) into high detail, which in return gave new insights in the general instructions Group Model Building entails.

2.2.1 The process

Ouborg (2015) has created a large set of BPMN (Weske, 2010) and ORM (Paulussen and Van der Weide, 2007) models. The combination of those two languages provided the possibility to create preconditions and postconditions to every part of the process. This in return gave insight in "what information was necessary" and "what information is produced" for every (sub)step of the process (Ouborg, 2015). This section briefly explains the findings of Ouborg (2015) by giving an overview of the models created. In the end this is translated into a list of concrete steps that explain basic Group Model Building.

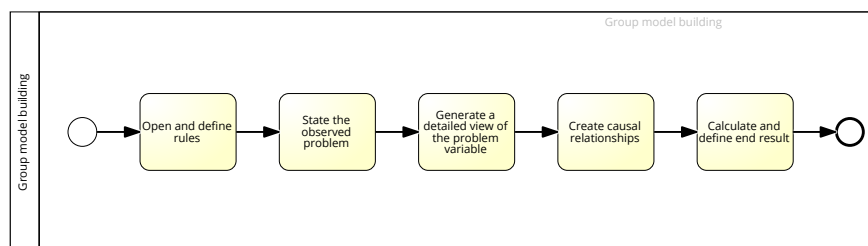


FIGURE 2.3: Overview of the basic Group Model Building process (Ouborg, 2015).

Figure 2.3 gives an overview of the basic Group Model Building process. One can see that it contains five basic steps (Ouborg, 2015):

1. *Open and define rules*: In this first step, the rules and regulations of Group Model Building are explained. For instance, the steps were explained, when it is allowed to talk to each other, when it is not. Also, in this step the setting of the facilitation is being checked, as people should be sitting in an U-shape (like the requirement for Nominal Group Technique). Ouborg (2015) concluded that this part of the process is highly dependent on the knowledge of the participants involved and is therefore not further formalized. It is up to the facilitator to determine the correct process. What is formalized was the information that is at least present after this step is being performed, namely: We know there is a set of participants and a facilitator and the participants are aware of the purpose, process, the rules and regulation of Group Model Building.
2. *State the observed problem*: In this step, the facilitator asks the participants what the variable of interest is. In most of the cases this is known prior to the session.

This is also preferable, as discussion about this variable and what it means can escalate into a general brainstorm (Ouborg, 2015). It is up to the facilitator to decide how to handle this, but should always be in the style of a process consultant if possible.

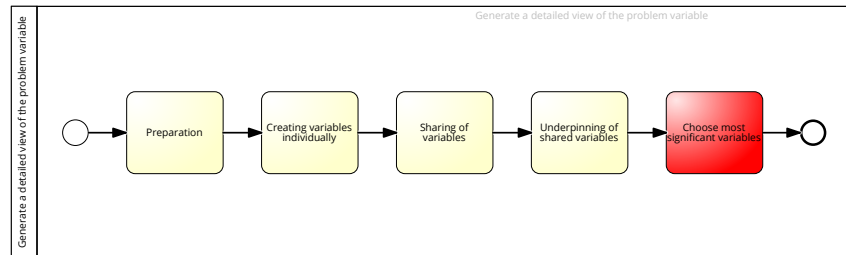


FIGURE 2.4: Overview of the Generate a detailed view of the problem variable process (Ouborg, 2015).

3. *Generate a detailed view of the problem variable*: Figure 2.4 shows the basic overview of this process. This generally represent the Nominal Group Technique part of Group Model Building and should be followed strictly according to these steps. Given that these were already explained in Section 2.1.2, we will not explain them here again due to readability.
4. *Create causal relationships*: After a detailed view of the problem variable is generated, feedback loops are being found. This happens by connecting step-by-step all the variables into a causal loop diagram. This diagram will then give insight in the feedback loops. Figure 2.5 shows the basic process of this step. These two steps

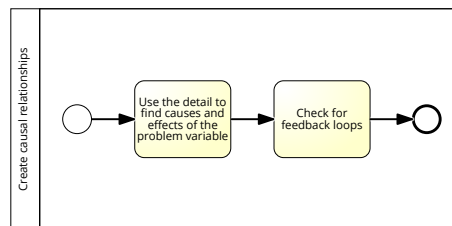


FIGURE 2.5: Overview of the Create causal relationships process (Ouborg, 2015).

in themselves contain a couple of sub-steps. Due to readability we will not give the BPMN-models here, but briefly give the steps which are as follows (Ouborg, 2015):

Use the detail to find causes and effects of the problem variable

- (a) First the participants are being asked to name a variable that can be seen as a cause or as an effect of the problem variable.⁶
- (b) After a variable is chosen, it is connected by the participant who came up with it to the model with a relation type, which can be positive or negative. A relation can also be unknown but this is highly unwanted, and it is better to leave the variable out if its relation to the other variable is unknown.

⁶In regular Group Model Building, the session will start with finding causes and when they are ready will continue with finding the effects of the problem variable (Vennix, 1996) (Rouwette and Franco, 2014).

- (c) After the connection is made by a participant, this connection is debated by all the participants.
- (d) Last, it is chosen by the facilitator (after receiving the input from the participants) to leave the connected variable the way it is, to park it, or to change its position and relation to the new variable.
- (e) After all these steps, the process goes back to a.

Check for feedback loops

- (a) The facilitator suggests a feedback loop to the participants. This is done based on his own experience of System Dynamics. Of course this choice must also be explained to the group.
 - (b) The participants now discuss if they agree.
 - (c) If they agree, it is written down as a potential feedback loop. Otherwise it can be discussed to park the feedback loop and come back to it later, change it, or keep it in the end the way it is.
 - (d) After this is done, the process returns to a.
5. *Calculate and define end result:* For many Group Model Building sessions a causal loop diagram would be its main product (Ouborg, 2015). But if a more quantitative approach is being chosen, the diagrams can be simulated with actual figures in order to determine when the escalating effect of a Messy Problem arise (Vennix, 1996). Figure 2.6 shows the basic parts of this last step.

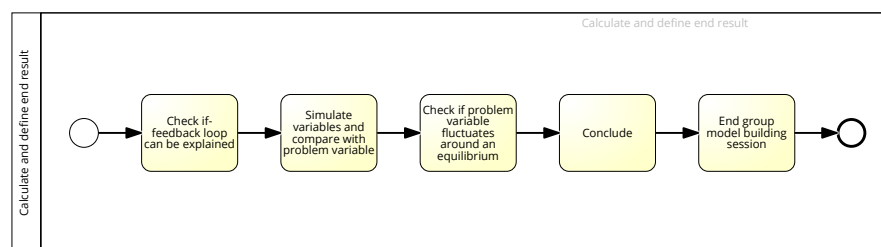


FIGURE 2.6: Overview of the Calculate and define end result process (Ouborg, 2015).

The five steps of this subprocess of Group Model Building are explained next (Ouborg, 2015):

- (a) *Check if feedback loop can be explained* In order to come to a solution, one should at least be able to explain the behavior of the feedback loop according to the causal loop diagram. Then an escalating effect of one of the variables can be explained by the nature of the other variables, and adjustments (and therefore potential solutions) can be made.
- (b) *Simulate variables and compare with problem variable* If feedback loops do not generate enough understanding of the behavior of the Messy Problem, they can be simulated (the quantitative approach of Group Model Building). A simulation will be made with every variable and the problem variable (variable of interest). This will show potential interesting behavior which can be a source for a better understanding.

- (c) *Check if problem variable fluctuates around an equilibrium* In the end, the ideal can be that the problem variable fluctuates around an equilibrium in the simulations. This means that no matter what the values of all the other variables are, there is no escalating behavior anymore. This gives in a quantitative research the solution to a Messy Problem. Adjust some of the variables in the simulation (which are strict) until balance has been accomplished. Sometimes escalating behavior might be the wanted result (for instance, when profit is stable and the simulation must give insight in how to increase its value). In that case the opposite should be done.
- (d) *Conclude* Not all sessions can be concluded with a solution to the Messy Problem, but the extra insights that have been created can be of large impact on the process of finding a solution. If the Group Model Building session is of a qualitative nature, then the previous two steps are skipped. A conclusion is most of the time a report written by the facilitator which entails the variables that have been found and the causal loop diagram that has been created. Also variables that have been parked or those who are not clear are explained in the conclusion document.
- (e) *End Group Model Building session* This last part might be one of the most important ones. As a facilitator, you should explain what the results mean and what can be done next. In the end, the participants should be confident that the causal loop diagram is a sound explanation of the behavior and should also be confident that they will manage to come to a solution in the end (or are at least closer to a solution).

These steps conclude the formalized processes of Group Model Building by Ouborg (2015). In his *Further Research* chapter, he states that these do not represent the Group Model Building technique fully, due to a couple of things (Ouborg, 2015):

- *The communication*: During the formalization process, the actions of the facilitator and of the participants were formally described. Although this gives great insight in the steps that are being taken, the communication between these two basic actors are not formalized and clarified. This was left out due to the fact that communication of these types of session, would be way too complex and dependent of the skills and style of the participants and the facilitator (Ouborg, 2015).
- *The coordination*: Not only what is communication but also which of the participants may communicate when is left out. For instance, if the Nominal Group Technique has finished and participants are allowed to name a variable they want to connect to the variable of interest, which participant is allowed to speak first?

This negligence of Ouborg (2015) is somewhat resolved in this thesis, given the fact that we let go the notion that everything has to be formalized one hundred percent. The *Facilitation principles* of Section 2.1.2 somewhat structures the universe of discourse when it comes to the facilitation of Group Model Building. Therefore, we now see these principles as part of the process of Group Model Building.

2.2.2 Criticism

Although a blueprint was created, this eventually did not mean that all information to create a digitized form of Group Model Building was extracted. In his *Further Research*

paragraph, Ouborg (2015) acknowledges that the role of the facilitator was mostly omitted during his research. One can argue that the role of the facilitator is hard to formalize in a set of sequential steps. A facilitator in general does not primarily act, but reacts to certain events (Schein, 1990, 1997). The quantity of these events can be unfeasible high. Normally, a facilitator reacts to events based on a number of principles, which together with his personal senses forms the basis of his reaction. To digitize a facilitator, a complex AI mechanism is necessary. Complex AI mechanisms are not expressible via the use of sequential business process modeling and data modeling, which would simply result in too many models.

Other critique on the work of Ouborg (2015) was that the sequential approach in general did not represent the core nature of Group Model Building. Group Model Building is not a simple cooking recipe that can be followed in order to gather unbiased and creative ideas. Its process is more of an agile nature, guided by a facilitator who determines the correct next step by sensing the current atmosphere of the process. A sequential approach could lead to more social loafing and free riding (see Section 2.1.2) which Group Model Building tents to counter.

The criticism clearly shows that another approach is required. Sequential formalization of Group Model Building is not the answer to create a reliable blue print for a distributed tool. This not only is concluded from literature, but will also be tested with a tool that represents the sequential approach (see Chapter 4). Besides the formal approach, other approaches have been created. The more agile *Dialogue Games* approach by Hoppenbrouwers and Rouwette (2012), will be elaborated in Section 2.4.

2.3 Basic requirements for Group Model Building

If we take into account all the literature research so far, we can formulate a couple of lists which represents the requirements of regular Group Model Building. These can be seen as features that should at least be present in an implementation for Group Model Building, in order to call it Group Model Building. However, these requirements must not be viewed as quantitative and fully formal, but more in a qualitative "should be somewhat present" way. In the end, every Group Model Building session is different due to its content and the communication style of the participants in question.

The requirements are not fully explained here due to readability, but there is referenced in which part of the literature an explanation can be found. Furthermore, we separated the list in a matter of process related requirements and process guidance related requirements. Both are evenly important for regular Group Model Building, but when creating a digital Distributed Group Model Building approach, a separation can be useful if one is only to digitize the process or both the process and the process guidance (facilitation) aspect of Group Model Building.

Group Model Building Process Requirements

Table 2.1 gives the basic requirements for a Group Model Building process.

#0	Basic Group Model Building steps	(Ouborg, 2015) and Section 2.2.2
Group model consists of the following steps:		
<ol style="list-style-type: none"> 1. Open and define rules 2. State the observed problem 3. Generate a detailed view of the problem variable 4. Create causal relationships 5. Calculate (or simulate) and define end result 		

TABLE 2.1: Requirement 0: Basic Group Model Building steps.

Table 2.2 shows the detailed requirements for the first step of Group Model Building, namely *Open and define rules*.

#1	Basic Open and define rules actions	(Ouborg, 2015) and Section 2.2.2
The Open and define rule step of Group Model Building, must have the following actions (order may differ):		
<ul style="list-style-type: none"> • Participants are welcomed • Explain rules and regulations • Make sure all the facilitation is in order (sitting in U-shape, pen and paper, whiteboard or general information screen) 		

TABLE 2.2: Requirement 1: Basic Open and define rules actions.

Table 2.3 shows the detailed requirements for the second step of Group Model Building, namely *State the observed problem*.

#2	Basic State the observed problem actions	(Ouborg, 2015) and Section 2.2.2
<p>The State the observed problem step of Group Model Building must contain at least the following actions (order may differ):</p> <ul style="list-style-type: none"> • Facilitator asks for variable of interest. Or, • facilitator gives a predetermined variable of interest 		

TABLE 2.3: Requirement 2: Basic State the observed problem actions.

Table 2.4 shows the detailed requirements for the third step of Group Model Building, namely *Generate a detailed view of the problem variable*.

#3	Basic Generate a detailed view of the problem variable steps	(Ouborg, 2015) and Section 2.2.2
<p>The Generate a detailed view of the problem variable contains at least the following steps according to the Nominal Group Technique:</p> <ol style="list-style-type: none"> 1. Preparation 2. Creating variables individually 3. Sharing of variables 4. The underpinning (explaining) of shared variables 		

TABLE 2.4: Requirement 3: Basic Generate a detailed view of the problem variable steps.

Table 2.5 shows the detailed requirements for the fourth step of Group Model Building, namely *Create causal relationships*.

#4	Basic Create causal relationships steps	(Ouborg, 2015) and Section 2.2.2
<p>The Create causal relationships step of Group Model Building must contain the following sub-steps:</p> <ol style="list-style-type: none"> 1. Use the detail to find causes and effects of the problem variable 2. Check for feedback loops 		

TABLE 2.5: Requirement 4: Basic Create causal relationships steps.

Table 2.6 shows the detailed requirements for the fifth and last step of Group Model Building, namely *Calculate and define end result*.

#5	Basic Calculate and define end result steps	(Ouborg, 2015) and Section 2.2.2
<p>The Calculate and define end result step of Group Model Building must contain the following sub-steps:</p> <ol style="list-style-type: none"> 1. Check if feedback loop can be explained 2. Simulate variables and compare with problem variable (in case of a quantitative Group Model Building session) 3. Check if problem variable fluctuates around an equilibrium (in case of a quantitative Group Model Building session) 4. Conclude 5. End Group Model Building session 		

TABLE 2.6: Requirement 5: Basic Calculate and define end result steps.

Table 2.7 shows the detailed view of the first part of *Generate a detailed view of the problem variable*, namely *Preparation*.

#3a	Preparation of Generate a detailed view of the problem variable steps	(Ouborg, 2015) and Section 2.2.2
<p>The Preparation sub-step of The Generate a detailed view of the problem variable step, contains the following actions:</p> <ol style="list-style-type: none"> 1. Tell that the group is most important here 2. Tell that the contributions of the individual are of value 3. Explain the ways of Nominal Group Technique 4. Present the nominal question (Variable of interest) 5. Explain that ideas should be written in short variable names 6. Make clear that there must be worked silently 		

TABLE 2.7: Requirement 3a: Preparation of Generate a detailed view of the problem variable steps.

Table 2.8 shows the detailed requirements of the fourth step of *Generate a detailed view of the problem variable*, namely *Underpinning of shared variables*. The second and the third step are not translated into detailed requirements, as these were already described enough by the general requirements of which they are part of.

#3b	Underpinning of shared variables of Generate a detailed view of the problem variable steps	(Ouborg, 2015) and Section 2.2.2
<p>The Underpinning of shared variables sub-step of The Generate a detailed view of the problem variable step, contains the following actions (order may differ):</p> <ul style="list-style-type: none"> • Choose a variable • Determine if it is understood • Ask for explanation if not understand • Explain variable if possible 		

TABLE 2.8: Requirement 3b: Underpinning of shared variables of Generate a detailed view of the problem variable steps.

Table 2.9 represent the detailed requirements of the first step of *Create causal relationships*, namely *Use the detail to find causes and effects of the problem variable*.

#4a	Use the detail to find causes and effects of the problem variable of the Create causal relationships steps	(Ouborg, 2015) and Section 2.2.2
<p>The Use the detail to find causes and effects of the problem variable of Create causal relationships steps contains the following actions (order may differ):</p> <ul style="list-style-type: none"> • Ask a participant to name a variable that can be seen as a cause of the problem variable • Choose a variable • Link the variable to another variable as a cause or effect with a relation type + - or ? • Discuss the linked variable if questions arise • Process the variable after the discussion 		

TABLE 2.9: Requirement 4a: Use the detail to find causes and effects of the problem variable of the Create causal relationships steps.

Table 2.10 represent the detailed requirements of the second step of *Create causal relationships*, namely *Check for feedback loops*.

#4b	Check for feedback loops of the Create causal relationships steps	(Ouborg, 2015) and Section 2.2.2
<p>The Check for feedback loops of Create causal relationships steps contains the following actions (order may differ):</p> <ul style="list-style-type: none"> • Suggest feedback loop, declare it to be a positive or negative one, and ask participants to comment • Discuss when disagreement takes place or if participants do not understand the feedback loop suggested • Process the feedback loop according to the result of the discussion 		

TABLE 2.10: Requirement 4b: Check for feedback loops of the Create causal relationships steps.

Group Model Building Process Guidance Requirements

Table 2.11 represent the basic requirements for a facilitator of Group Model Building.

#6	Facilitation requirements	(Schein, 1990, 1997) and Section 2.1.2
<p>A facilitator must act according one of the following three styles, always starting with the first:</p> <ul style="list-style-type: none"> • Process consultation • Playing doctor • Providing expert information 		

TABLE 2.11: Requirement 6: Facilitation requirements.

Table 2.12 represent the detailed requirements of the most use style of facilitation, namely *process consultation*.

#6a	Process consultation requirements	(Schein, 1990, 1997) and Section 2.1.2
<p>A process consultant must act according to the following principles:</p> <ul style="list-style-type: none"> • Always be helpful • Always deal with reality • Access your ignorance • Everything you do is an intervention • The client owns the problem and the solution • Go with the flow, but seize targets of opportunity • Be prepared for surprises, and learn from them • Share the "problem" 		

TABLE 2.12: Requirement 6a: Facilitation requirements.

These lists of requirements now form the answer to the first research question, the Design Question: *What are the basic requirements for Group Model Building?*. The requirements tables explain what is necessary for an implementation of a collaboration technique in order to call it Group Model Building (be they analogue or digital). We have seen that these requirements consists of two basic parts and a distinction should be made between what parts of Group Model Building are digitized (when one wants to create a digital approach). It can be so that only a small part of the requirements are present in the digitized form and the rest still is in its old fashion analogue form. Although this might be the case, in the end all the requirements have to be identifiable inside the tool that is being used, with its corresponding support (be they in the form of facilitation).

Chapter 3 talks about the tool that is being used to digitize our approach of Group Model Building in order to create a distributed approach to it. This is done by explaining

the framework in which it is build, namely Slack. Slack is a widely used platform for developers and is growing fast in users. Therefore this potentially creates an easy to use approach to Distributed Group Model Building, which can be implemented at a low threshold. But before we can do that, we create a combined approach to Group Model Building in which we find a solution to the sequential pitfall of Ouborg (2015). This is done in next section.

2.4 The Dialogue Games approach

As was mentioned in the introduction (Chapter 1), the other source of inspiration for this research was the paper by Hoppenbrouwers and Rouwette (2012). In their research, they tried to create a digitized implementation for Group Model Building by linking it to *Dialogue Games*. Their result was a pioneering digitized form of Group Model Building that could be used in a distributed manner. For this approach they used InterLoc, which is an online chatting tool for Dialogue Games.

In their research (which was also conducted according to the Design Science methodology (Hevner et al., 2004)), Hoppenbrouwers and Rouwette (2012) used a couple of principles. For instance, they acknowledge that *"the guidelines and rules on which our approach are founded are less strict than stepwise processes as typically captures in traditional, flowchart-like procedures"* (Hoppenbrouwers and Rouwette, 2012) which is the opposite approach of Ouborg (2015). Their choice is logical since exceptions on the stepwise processes will likely occur and cannot be neglected (van Gils and Vojevodina, 2006). Just by providing the boundaries, or no-go-area's, the users of the technique will know what to do, without doing something illegal. Game Theory, on which Dialogue Games are based, provide a framework for describing these boundaries with basic guidelines and rules (Hoppenbrouwers and Rouwette, 2012).

Although Hoppenbrouwers and Rouwette (2012) created a digitized form of Group Model Building, it was not their intent to improve collaborative modeling. Their implementation is just an example of what is possible with this Dialogue Games approach (Hoppenbrouwers and Rouwette, 2012). However, they formulated a couple of potential benefits of this approach (Hoppenbrouwers and Rouwette, 2012):

- *Games are coherent and systematic like scripts* Scripts in general do not only tell you what to do, but also can give transparency into the whole process. Participants understand better what is expected of them and will act more naturally. The use of scripts in Group Model Building has been done before (Hovmand, Andersen, Rouwette, Richardson, Rux, and Calhoun, 2012), where is concluded that the use of scripts *"makes it possible to more effectively engage a wider and more diverse set of stakeholders where conflict and coordination issues may be major barriers to solving some problem in a system"* (Hovmand et al., 2012). This generally is due the fact that planning can be done more explicit (Hovmand et al., 2012). The use of scripts in games is therefore a potential benefit for a Group Model Building implementation.
- *Chats are more easy to analyze* Hoppenbrouwers and Rouwette (2012) where inspired by the use of an online chat tool by the fact that they encounter the transcribing of analogue collaboration to be lots of works. This could easily be overcome if online chat tools were used , but only if these tools support some form of

automated logging (Hoppenbrouwers and Rouwette, 2012). The produced log file can then be analyzed automatically (e.g. with the use of algorithms).

- *Game create a platform which makes it more easy to connect to other people* In regular Group Model Building, all the relations go via the facilitator. In contrary, games make it possible to isolate some of the actions which in some cases makes the facilitator unnecessary (Hoppenbrouwers and Rouwette, 2012).
- *More or less strict rules can be formulated in which freedom can be given* In contrast, the approach of Ouborg (2015) used rules in terms of how less freedom could be given. For instance, a BPMN model states what the exact process is. But from that exact process, under certain circumstances, can be deviated if the facilitator finds that necessary. The rules for freedom like Hoppenbrouwers and Rouwette (2012) is using, the facilitator and participants are allowed to do anything they want, provided that certain rules are obeyed. Although subtle, this difference in approach create the foundries for either a formalized or an agile approach to Group Model Building, or any other collaboration tool for that matter.

With the notion of these benefits, Hoppenbrouwers and Rouwette (2012) created a Dialogue Games approach of Group Model Building. In order to derive the rules necessary, they gathered Group Model Building scripts and performed the FoCon method (Hoppenbrouwers and Wilmont, 2010) to those scripts. The *"FoCon analysis is conceived as an aid in identifying the foci⁷ relevant to specific types of modelling, including all the steps taken in its operational (i.e. real, applied, enacted) modeling process."* (Hoppenbrouwers and Wilmont, 2010). This resulted in a *"more system-oriented re-interpretation of the scripts in terms of phases in the interaction, with distinctive conceptual input and output elements (e.g. 'variable', 'cause', 'loop')"* (Hoppenbrouwers and Rouwette, 2012).

After these scripts were analyzed with the FoCon method, they were translated into *openers*. The whole purpose of Dialogue Games is to provide a limited set of directions by using rules. *Openers* can be used as rules since they limit the sentences that can be produced. The chat tool that supports openers is *InterLoc* (Ravenscroft and McAlister, 2006), which was used by Hoppenbrouwers and Rouwette (2012). Apart from the openers, *InterLoc* is a general chat tool with the support of threads. These thread will automatically initiate depending on what opener has been used. Figure 2.7 shows the list of openers created by Hoppenbrouwers and Rouwette (2012).

With the *InterLoc* tool ready, Hoppenbrouwers and Rouwette (2012) performed a test in practice. All the users were separated and could not speak to each other other than by using *InterLoc*. The facilitator of the sessions was an experienced Group Model Building facilitator who besides *InterLoc*, also used *Vensim*⁸: a tool for modeling causal loop diagrams. The facilitator was building the model live, which was also visible to the participants. The participants themselves were students without Group Model Building experience.

⁷Which can be a *pragmatic focus* or a *semantic-syntactic focus* (Hoppenbrouwers and Rouwette, 2012).

⁸Site: vensim.com.

-
- Propose (only to be used by players)**
 - I propose the following IDEA:
 - I propose the following VARIABLE for the idea:
 - I propose the following IDEA as expressed through the following VARIABLE:
 - I propose the following CAUSE with its POLARITY [variable, +/-]:
 - I propose the following CONSEQUENCE with its POLARITY [variable, +/-]:
 - I propose that the polarity of this variable is [+/-]:
 - Ask (only to be used by players)**
 - I have a question:
 - I have a question about this proposition:
 - Argue (only to be used by players)**
 - I agree:
 - I disagree:
 - Accept / Reject (only to be used by players)**
 - I accept the proposition:
 - I reject the proposition:
 - Remark (only to be used by players)**
 - I would like to clarify this:
 - I have a remark:
 - Facilitator statements and questions (only to be used by facilitator)**
 - Instruction of the facilitator:
 - Directive of the facilitator:
 - This is the problem variable:
-

FIGURE 2.7: Openers for the InterLoc implementation of Group Model Building (Hoppenbrouwers and Rouwette, 2012).

After the test, Hoppenbrouwers and Rouwette (2012) concluded the following:

- The use of InterLoc/Dialogue games ”induces a more relaxed pace of interaction, allowing for reflection in between contributions.” This can of course highly impact the quality of the content that is created and in Group Model Building provide a better understanding of the Messy Problem.
- They could not conclude whether this platform would also worked with a less experienced facilitator, or a facilitator with little to no Group Model Building experience.

This concluded the research by Hoppenbrouwers and Rouwette (2012). Compared to Ouborg (2015) it is not clearly stated that one approach is better than the other, both have their benefits. Ouborg (2015) provided a far greater detail in the processes of Group Model Building than Hoppenbrouwers and Rouwette (2012), but Hoppenbrouwers and Rouwette (2012) in return provides a better understanding of the nuances collaborative modeling entails. Therefore, a combination of both theories combined with the extra research done in this thesis would provide an improved approach in creating a distributed form of Group Model Building. This combined approach is explained next.

2.5 The combined approach for Distributed Group Model Building

Basic Group Model Building methodology and the implementations of Ouborg (2015) and Hoppenbrouwers and Rouwette (2012), provide the basis for our approach for a Distributed Group Model Building application. Our ideal is that the implementation of Group Model Building described in this paper, is also usable for less experienced facilitators. We therefore react to the second conclusion of Hoppenbrouwers and Rouwette (2012) (see Section 2.4). Before we explain how are combined theory is constructed, we explain two additional theories that extends our theory so its foundries are more sound. These theories are those of *Scaffolding* (Oppl, 2016) and *ThinkLets* (Hoppenbrouwers and van Stokkum, 2013).

2.5.1 Scaffolding & ThinkLets

The following two theories explain two aspects that can be of great importance if one wants to have sound team collaboration. Besides the methodologies of Group Model Building and especially the one of facilitation, the theory of *Scaffolding* (Oppl, 2016) can give insight in how to manage information exchange from the facilitator to the participants.

Scaffolding is "*a temporary means of support that is present until the entity supported by scaffolds (here: a subject participating in conceptual modeling) can accomplish a given task herself.*" (Hoppenbrouwers and van Stokkum, 2013). For instance, when a facilitator understands how the Nominal Group Technique works and is able to explain this soundly, then there is no need for the participants to be informed otherwise about the Nominal Group Technique (assuming they will always understand the process after the explanation of the facilitator). The facilitator here is using Scaffolding (explaining) to give a scaffold (rules and regulations of Nominal Group Technique) to the participants.

Scaffolding can be categorized in different sub-types (Jumaat and Tasir, 2014) (Oppl, 2016):

- *Conceptual scaffolds* Helps to prioritize fundamental concepts to make a choice in what is important to learn.
- *Procedural scaffolds* Helps to find useful resources, such as tools and methods.
- *Strategic scaffolds* Helps in tackling learning problems by presenting alternative ways.
- *Metacognitive scaffolds* Helps in tackling learning problems by explaining what to think and how to elaborate.

This distinction can be used in helping to use the principle of scaffolding in Group Model Building. But before we describe this (see Section 2.5.2), we should mention that Scaffolding can be done in certain ways (Bulu and Pedersen, 2010) (Oppl, 2016):

- *Scaffolds provided by teachers* In our example, this would be the facilitator explaining the Nominal Group Technique.

- *Scaffolds provided in interactions among learning peers* If after the explanation of the facilitator a participant does not understand what is meant with the Nominal Group Technique, he or she can ask another participant to explain again. That other participant is now performing scaffolding.
- *Scaffolds as textual or graphical representations* When we give the rules and regulations of Group Model Building on a sheet of paper, it would be textual Scaffolding.
- *Technology-driven scaffolding* This would be possible in our approach, when Scaffolding is done by the tool itself (i.e. the tool explains the Nominal Group Technique on the fly).

How Scaffolding is used in this thesis is explained in the next section (Section 2.5.2).

The last piece of theory is that of *ThinkLets* (De Vreede and Briggs, 2005) (Hoppenbrouwers and van Stokkum, 2013), which will be explained briefly. *"A thinkLet is a named, packaged facilitation intervention that creates a predictable pattern of collaboration among people working together toward a goal. A thinkLet is meant to be the smallest unit of intellectual capital needed to allow a process designer to communicate the implementation of a facilitation intervention so that others can successfully reproduce the patterns of collaboration the intervention is meant to create."* (De Vreede and Briggs, 2005).

In other words, a thinkLet represents the smallest amount of information that is needed for a participant to allow to collaborate within a Group Model Building process. ThinkLets in our approach would represent the information the facilitator is equipped with to be able to facilitate a Group Model Building session. This can be inspired by notion of Scripts (Hovmand et al., 2012), which was also used by (Hoppenbrouwers and Rouwette, 2012) (and also explained in Section 2.4). ThinkLets (Scripts) will be used knowingly when the implementations of a distributed tool are defined. Linked to Scaffolding, thinkLets would be the most compressed form of scaffolds that enables Scaffolding. Both thinkLets and Scaffolding will come back in the next section, where we bring all the theory explained thus far together and translate it into our approach for a digitized form of Group Model Building.

2.5.2 Combined approach

Now that we have elaborated all the theory, we can describe what the theoretical foundation of our approach to Distributed Group Model Building will be. For readability and structure, we categorize the theoretical foundation in *process*, *facilitation*, and *information*. Combined they form the basis for the implementation using Slack (see Chapter 3). As was mentioned earlier in the introduction (Chapter 1), we will create two Slack implementations for practical testing, namely: the approach by Ouborg (2015), and the combined approach of this thesis. The reason for doing so is that the approach of Ouborg (2015) was never tested in practice which is not according to the Design Science approach. Only when this is completed, Ouborg (2015) forms a Design Science sound foundation for this thesis.

Combined approach: The Process

Our combined approach of Group Model Building will use the processes formalized in Ouborg (2015) as its foundation. That foundation then represents the ideal Group

Model Building session, in other words, one that is going fluently without trouble from begin to end. Given van Gils and Vojevodina (2006), exceptions should be expected. Therefore, we acknowledge that it must be allowed to jump between the last three phases of Group Model Building, but only when a process has already passed (so if you are in the *Generate a detailed view of the problem variable* for the first time, you are not allowed to jump to *Calculate and define end result* before executing *Create causal relationships*). This interpretation can be seen in Figure 2.8.

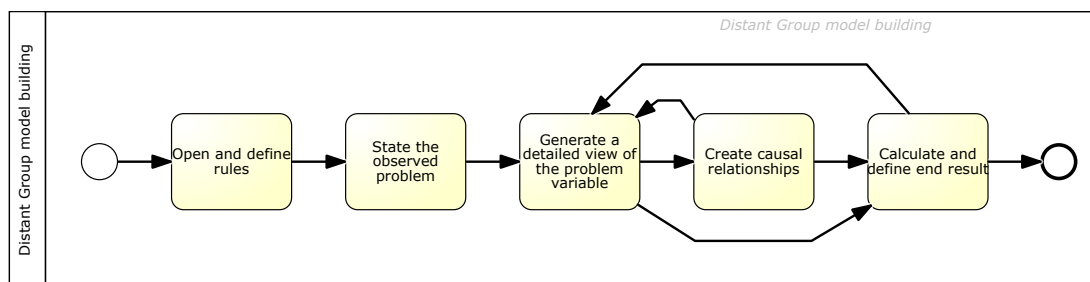


FIGURE 2.8: The main process of the combined approach of Group Model Building.

Another important notion is that the process is not guided by a sequential approach, but by using a Dialogue Games approach like in Hoppenbrouwers and Rouwette (2012). In many cases the choices that lay in front of the participants determine the process overall, instead of the facilitator who is controlling all the conversation. This creates complexity that is hard to model or to formalize. In addition, the platform on which the implementation is build also limits the possibilities of such an approach. Therefore, the choices that are available to the participants will be described in Chapter 4 where the implementation of Distributed Group Model Building is given.

One last remark to make here, is that the more agile approach here still must be seen as an exception of the ideal sequential flow of Group Model Building. The tool that is being created should always try to follow the sequential path as much as possible. This is due to one simple reason, namely that the jumping between steps would create too much debate freedom and disregard the fundamental principle of Group Model Building and System Dynamics that "Structure drives behavior" (Forrester, 1961, 1975) (Vennix, 1996) (Rouwette and Franco, 2014) (Ouborg, 2015).

Combined approach: The Facilitation

The facilitation in the combined approach of Group Model Building (given the tool that is being created) is a combination of *Scaffolds provided by teachers* and *Technology-driven scaffolding* (Bulu and Pedersen, 2010) (Oppl, 2016). The first represents the facilitator and the latter represents the tool. Our goal is to create a tool that would also enable facilitators without Group Model Building experience to facilitate a Group Model Building session. This is done in two ways: The tool will explain to the person in question what can and what must be done by a facilitator, and the tool explains certain requirements to the participants of Group Model Building. The combination of these two create a technology-driven Scaffolding which contains a thinkLet (De Vreede and Briggs, 2005) (Hoppenbrouwers and van Stokkum, 2013) for basic understanding of Group Model Building, without prior training.

Furthermore, the facilitation of the combined approach must follow the advised style of facilitation, namely that of *process consultation* (Schein, 1990, 1997) (Vennix, 1996)

(Rouwette and Franco, 2014). Herewith, we mean the facilitation by a human. The facilitation by the tool is primarily explaining the steps and ways of the process to follow and is therefore more in the form of *providing expert information* (Schein, 1990). For more details on these ways of facilitation, see Section 2.1.2.

Combined approach: The Information

Given our dialogue games implementation, we let us inspire by the *openers* created by Hoppenbrouwers and Rouwette (2012). During the session, participants will be able to chose one of these openers after which the tool takes over. The tool then provides basic information necessary to answer correctly or in a given format (according to the fundamentals of Group Model Building). Also the tool will provide the necessary information so participants and the facilitation understands what to do next. The tool in general acts as a sort of on-the-fly manual (or guide), that gives guidance during a Group Model Building session.

In addition to the information, one has to mention the data that is produced by the participants and the facilitator. To know what is at least required in order to be able to continue, we let us inspire by the preconditions and postconditions of Ouborg (2015). Within a chat, what is written provides the criteria to be able to say if these conditions are met. Therefore the chat itself is the largest source of information for the Group Model Building session. Not only the communication but also the (intermediate) results are present here. Therefore in our combined approach, there should always be logged what is being said by whom. Only than the information is sound for Distributed Group Model Building.

This concludes this literature research. The elaborated theory (in addition to Ouborg (2015)) and linking that to Group Model Building as well as the combined approach forms extended knowledge in general. According to Design Science (Hevner et al., 2004, Hevner, 2007), this completes the *Rigor Cycle* and extends the *Knowledge Base*. To complete the Design Science Research methodology, it now has to cope with the *Environment* by completing the *Relevance Cycle*. This can be done by creating the two implementations and testing these in practice (Field testing). The Field testing then forms the last part of this research and is described in Chapter 5 and Chapter 7.

Chapter 3

Using Slack

For the Distributed Group Model Building tool, Slack is being used. Slack is an online platform for collaboration and chatting, specified in supporting project teams.¹ It is gaining fast in users since its first deployment in 2013. While they hit a 100.000 users in 2014, in 2016 they already had 3 million users (The Atlantic, 2016), which translates into about 33.000 teams (Fast Company, 2017).

This chapter explains why Slack is chosen as the basis for a Distributed Group Model Building platform, by first explaining the features of Slack and how this benefits the Group Model Building tool. Furthermore, a basic explanation of the tool is given, by translating the literature of Chapter 2 into a design suggestion for the tool.

3.1 About Slack

Slack can be seen as a *Team communication platform* (TCP), which enables *Enterprise social media* (ESM) in an *Enterprise social network* (ESN). ESN is the use of the features of social media in an internal fashion (Anderson, 2016). As Anderson (2016) mentions: "*ESN tools provide functions that promote collaboration and communication between both individuals and groups. The use of ESN helps create a community within an organization just as the use of social media helps create community around organizations, business, hobbies and special interests. ESM enhances internal communication and social interaction within an organization.*" His conclusion is that Slack as an ESN tool provides "*the opportunity for organizations to take what works with external communication and apply it to internal communication*" (Anderson, 2016).

Slack can be used to support the internal communication of project teams in a general sense. While originally build to support software development (Lebeuf, Storey, and Zagalsky, 2017), it now has found its way into other project types (Perkel, 2016) (Marshall, Gamble, and Hale, 2016) (Perrin, 2015). Some now even prefer Slack over e-mail due to the fact that its signal-to-noise ratio is high (Perkel, 2016). This is because Slack has channels which define a certain topic boundary. This has as a result that messages are pre-ordered and tagged on importance. A lesser important non subject message, will probably be posted in *#random*, which is a predefined channel for fun, junk or other unrelated conversations.

¹Slack: slack.com

Besides its spam prevention and message overview, Slack has also been claimed to help in building consensus (Perkel, 2016). In their research, Perkel (2016) talks about how at MIT the building of graphs in preparation for a Nature publication went way faster and more easy when Slack was used. This was due to the immediate feedback that could be delivered and narrow scope discussions that would follow on how to do it otherwise. But, Perkel (2016) not only praises Slack. Given that Slack is not an open source type of software, privacy concerns arise. MatterMost (an open source variant of Slack) is suggested as an alternative.

The most interesting part of Slack is its support for chat bots (Marshall et al., 2016) (Lebeuf et al., 2017) (Perrin, 2015). A chat bot listens to the conversations and reacts when necessary. Therefore they can support the communication within teams, or be an administrative aid. Chat bots can potentially reduce friction within project teams (Lebeuf et al., 2017) and therefore enhance the more unemotional impartial communication, which in most settings is preferred (Marshall et al., 2016).

How this all comes together for our Distributed Group Model Building tool is explained next. But before we can do so, we have to elaborate the basic functions of Slack to explain its potential (Section 3.1.1). After that has been accomplished a translation into the Group Model Building context can be made (Section 3.2).

3.1.1 Basic functions

In this section, the basic functionality of Slack is elaborated. Slack is not a standard chat application like Skype or Whatsapp. It is more build around all the administrative tasks you would encounter when working in project teams. Therefore one can state that it is an integration of not only a chat functionality but also a file sharing and team management functionality. To accomplish this, Slack has a couple of features, namely: *Channels*, *Personal messages*, *Threads*, and *Bots*.

Slack denotes different chat sessions as channels. Channels normally represent different parts of a project. For instance, within a software company you have a channel "development", "design", and "testing". You can add users to channels and therefore create sub-teams within Slack. A manager who is interested in monitoring all the activities is able to enter all the channels.

Channels can be public or private. The difference between the two is that in a private channel, only a channel member can see the channels content. In a public channel the content is visible for everyone who is part of the Slack Group (the collection of Channels). Only members of channels can add content to the channels.

Slack also supports bilateral conversations between two users. This is called a personal message. If users do not see the benefit of conversation which is visible for a whole group, they should use this personal messaging service.

To structure a channel, but more importantly to keep it clean, one can react directly to a single message using a thread system. Slack will only say that a reply to a message has been made and leave it up to the user to make it visible. Therefore different subjects can be kept ordered and it is prevented that different subjects are crossing each other in a linear chat which can create misunderstandings.

The most interesting part of Slack is its adaptability using Bots or Apps. A bot is functioning as a virtual user who listens to the channel in which he is invited. If a message is posted, the bot will process this message, and react if necessary. In practice, a bot is commonly used as an administrative helping hand. Teams can mention the bot and asks specific questions in which the bot reacts with the information necessary.

An example of a bot is a *"TODO-list bot"*. Teams can mention the bot and ask it to add a task to a list. The bot will ask in return if the task has a deadline. If so the bot can automatically give updates on approaching deadlines of tasks. In this case the bot functions as a remembering aid.

3.1.2 A Slack Bot

A Slack bot is nothing more than a listener and reactor of a chat session. A bot can be added to the channel and acts as a normal user. Only in the background this "user" is automated. Figure 3.1 shows the basic process of a Slack Bot. Every time a message is posted in a channel, the bot looks at this message. If the message contains information which triggers an action, the bot will react appropriately. This in return is visible as a posted message in the chat.

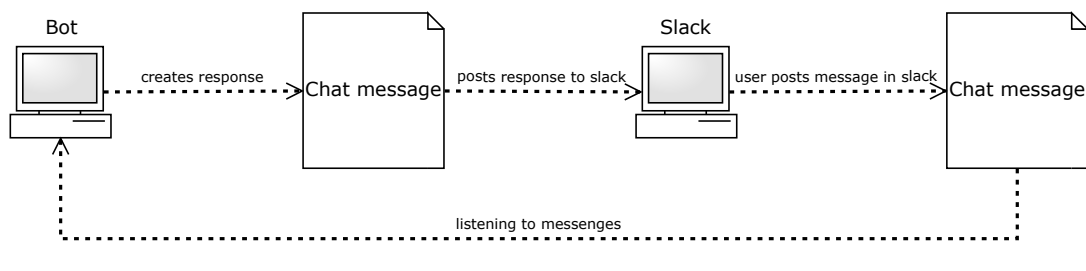


FIGURE 3.1: Basic principle of a Slack Bot.

The most interesting part is therefore how the Bot is programmed. There is almost unlimited potential in this approach, but requires a lot of AI skills when done properly. The benefit of a Dialogue Games approach is that the bot in question should only be programmed to follow the few steps and procedures defines within the Dialogue Games method.

3.2 Translating the literature into a Slack App

Given the flexibility of Slack, its features, and its popularity, it is chosen as the platform to develop a Distributed Group Model Building tool. For Distributed Group Model Building a Slack App is developed, which contains a Slack Bot that is presenting itself as a guidance for the Group Model Building process. In the next sections, the context and requirements described in Chapter 2 will be used as a starting point of the design of this Slack App. The actual functional design is given in Chapter 4 and Chapter 6.

3.2.1 Slack Bot role within Group Model Building

The first question that needs to be asked is what role the Slack Bot plays in the application. On the one hand it can act as the facilitator. In this case the bot fully controls who is allowed to communicate with whom and what results are being created. In the end this would be the ideal position of the bot, because every team could use the Group Model Building technique without the need for an experienced facilitator.

However, an approach where the Group Model Building bot is in full control is unfeasible to realize due to its complexity as was mentioned earlier. So, can there be a middle way? Given that we ideally want to create a tool that is also usable for those who are inexperienced in Group Model Building, this should be a "must"² requirement. One can argue that inexperience in Group Model Building does not necessary have to mean that one is inexperienced in facilitation or chairing a meeting in general. Therefore, the Bot can be used as a guide for the Group Model Building process without fully control the conversations that take place.

A middle way of the Bot being a guide is easy to implement, since the bot does not have to understand *what* is being said. It only has to understand when to go to a different phase in order to give information of what should be done next in the process. It is up to the facilitator of the session, which can be a leading team member, to steer the bot and to fill in the contents of these processes. How the role of the facilitator takes place exactly is mentioned in the next section (3.2.2).

The Bot can also provide a structuring function, which is extracted from the Dialogue Games principle. For instance when a variable is shared with the group to give comment on, instead of letting the users freely write their opinion, the Bot provides a number of possible buttons to click. These buttons can for instance be:

- *out of scope* Can be chosen if a user thinks the chosen variable is not within the boundaries of the discussion.
- *ask a question* Can be chosen when a user does not (fully) understand the meaning of the variable.
- *suggest* When a user can suggest another variable or a modified one.

After such a button is pressed, the Bot can guide the user in a bilateral conversation by asking specific questions. In this way, the commentary of a user on the variable will be written in a strict predefined format. The buttons and the question-answer principle can be seen as an implementation of the Dialogue Game principle. The formatted commentaries prevent commentary from being influenced too much by emotion or politics, which benefits the session overall and is in the end one of the main goal of Group Model Building.

3.2.2 Facilitators role within Slack

Given that the Group Model Building Bot now acts only as a guide for the Group Model Building process and as a Dialogue Games manager, what should be the role of the group's facilitator? What is his main task? And, what are the required skills?

²Derived from the MoSCoW prioritization system (Achimugu, Selamat, Ibrahim, and Mahrin, 2014).

The role of the facilitator is mainly to be a leading user. He or she functions just as a normal user of the tool, but promote him-/herself in taking the lead. This means he or she will give the variable of question and functions as an administrator of the tool.

The main task that this leading user as a facilitator has to do are the following:

- He or she should give the main variable at the beginning. This does not mean he or she decides what that variable is, but the leading user will only administrate the process of defining the main variable.
- Gives general instructions to the Bot. For instance, when the group is ready to proceed to the next step of Group Model Building, the leading user can give the Bot the instructions to proceed in which the Bot will go to the next step of Group Model Building.
- The leading user will perform all adding and adjusting of variable tasks that have to be done during the processes of Group Model Building. For instance, after a Dialogue Games session it is decided within the group to adjust a variable, only the leading user can do this. This is done to maintain control and a clean structure of the session.
- The leading user should facilitate. This means it has to act like suggested in Chapter 2: be a process consultant (Schein, 1987, 1990).

The required skills are therefore in general one of a basic mediator that is aware of the politics and can position itself in a neutral stance in the session. Being neutral is the most important part (when creating consensus), therefore the processes of the Bot should enable the leading user to be just a process consultant, and should therefore prevent him or her for primarily pushing forward his or her own ideas, but more importantly the bot should create an atmosphere in which the leading user can act in a way in which also the others think he or she is neutral within the discussion.

To conclude, the Slack Distributed Group Model Building App should contain a Bot that guides the process of Group Model Building. This enables users without prior knowledge of Group Model Building to use the technique. It is still required to have a leading user acting as a process consultant (facilitator), given that the Bot guides the user in being so. In the end there is a fair balance between feasible implementation and the simplification of the use of Group Model Building in a distributed fashion.

Chapter 4

The Sequential Group Model Building Bot

Previous chapters explained the theoretical background of Group Model Building and its general context. Studies by Ouborg (2015) and Hoppenbrouwers and Rouwette (2012) were examined and used as inspiration for a combined approach. A digitized form of the combined approach of Group Model Building is the main product of the research of this thesis. But, before we can create this soundly according to the Design Science methodology (Hevner, 2007), we have to create a digitized version of the approach of Ouborg (2015).

A digitized version of Ouborg (2015) will complete the Design Science cycle of that research, which makes it a correct foundation for this research. Also, given that the combined approach relies on the formal processes generated in Ouborg (2015), digitizing that approach gives a nice basis for the digitized version of the combined approach.

4.1 Technical choices

In order to create the digitized form of Group Model Building, a couple of choices have been made. First (as was already mentioned in Chapter 3), Slack is chosen as the platform to develop the tool on. Luckily, Slack has great support for building your own extension to the platform, as is described in their API documentation.¹ But, given the large amount of communication that takes place and the complexity of the content, the use of just the Slack API would be infeasible. The infeasibility lays in the fact that a lot of repetitive steps have to be taken, which would increase the number of lines of code that has to be produced. Thus, a different solution has to be found.

In order to keep the number of lines of code low, we have found a solution that acts as a layer between the Slack API and our own Bot. This layer is called BotKit² and was created by the Howdy team. It provides a simplification of functions that interact with the Slack API. This enables the possibility to post a message to a Slack channel, by just firing one function (instead of the six functions Slack requires). The BotKit

¹Site: <https://api.slack.com/>

²Site: <https://www.botkit.ai/>

platform uses the NodeJS language and framework to program, and must run with your own implementation on a dedicated server which is in return connected to Slack. Figure 4.1 shows the basic principle of this technical architecture.

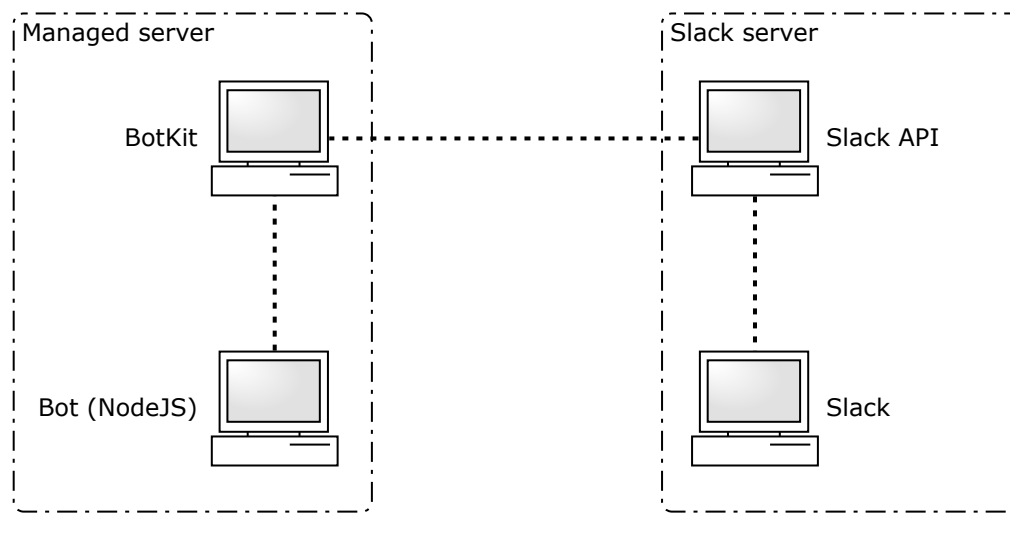


FIGURE 4.1: The basic architecture of the bot and Slack.

These technical details now form the basis for our implementation. The further steps basically consisted of programming step-by-step the formalized process of Ouborg (2015). During this creative process a lot of challenges arose, some things were not feasible to program which meant alternatives had to be sought. Furthermore during the creation of the tool, Dialogue Games automatically sneaked into the end result. This was due to the fact that Slack support *buttons* to chose further action. These buttons in some way are equal to the openers of the InterLoc system (Ravenscroft and McAlister, 2006), which automatically created a Dialogue Games thinking.

Therefore, the implementation of which the results are described in the next section where a combination of the purely sequential approach and a bit of Dialogue Games. This is still however a purely digitization of Ouborg (2015) because no agile approaches were implemented. It only represented a Group Model Building session in its ideal situation (from begin to end without exceptions). How Dialogue Games were implemented exactly, is explained on the fly.

4.2 The sequential Distributed Group Model Building tool

This section describes how the implementation of Ouborg (2015) is being made. Due to size and readability, we will not explain with figures how this implementation looks like. We only describe certain aspects of the tool and choices that have been made given the features of Slack. Chapter 6 shows our final tool with descriptive figures. To structure the description, we will act like we are performing a Group Model Building session using Slack and go step-by-step through the tool.

4.2.1 Open and define rules

To start a Group Model Building session, all the participants should be active on Slack and should all be invited to the same channel. In our example we chose the channel to be *#masterthesisdemo*. If all the participants are active and in the same channel, one of the participants can ask for the Group Model Building Bot to activate by typing `\gmb` in *#masterthesisdemo*. This will activate the bot, and the bot will welcome you. Furthermore the bot explains what his function is and asks who of the group (participants in the channel) will act as facilitator. One of the participants may respond and will act as the facilitator from now on. The bot will tell the group that this person from now on is the facilitator, which is called *leading user* in Slack. The bot automatically continues to the next phase, see next Section.

4.2.2 State the observed problem

In the next phase, the Bot will tell the group that a central variable is needed and that this will be defined by the leading user (i.e. facilitator). Now a collaboration between the bot and the leading user is happening. In a separate bilateral chat, the bot asks the leading user for the variable. The leading user can just give the variable to the bot if that variable is already defined. In the case it is not, it is up to the leading user to ask the group for this variable. The smartest thing to do here would be to just give any variable, due to the next step.

If a variable is given, the Bot will present this variable to the group in *#masterthesisdemo*. It tells the leading user the general conversation is now continuing in *#masterthesisdemo* and if he or she wants to adjust the central variable, he or she can do this in the bilateral conversation between the bot and the leading user. In *#masterthesisdemo*, the bot now has posted the variable with a couple of options as buttons. These buttons form the first implementation of Dialogue Games and when pushed will create a thread where opinions will be posted. The buttons that can be chosen are as follows:

- **Okay** With this button, participants can easily make clear that they agree with the suggested central variable. This will create a thread message that states that *x agrees!* (*x* is the user who pushed the button).
- **Make a suggestion** This button can be chosen if a participant wants to suggest another variable or an adjusted one. This will trigger a bilateral between the that participant and the Group Model Building Bot in which the Bot asks questions so that the opinion of the user is according to a predefined format. After the questions have been answered, the bot posts a message to the thread of the central variable and states that *user x suggested y because z*.
- **This is out of scope** Just like *Make a suggestion* the Bot will start a bilateral conversation with the participant who pushed this button and will ask the participant why he or she thinks the current central variable is out of scope. The Bot then posts in the thread of the central variable that *user x thinks this variable is out of scope, because y*.
- **Make a remark** If a participant just has something to say about this variable, he or she can push this button. This will also trigger a bilateral conversation between

the Bot and the participant, and the Bot will ask what their remark is. After this has been answered, the Bot will post a message in the thread of the central variable, stating that *user x has a remark, namely: y*.

If after the participants have given their opinion, the leading user can choose to do two things: First, he or she can choose to adjust the central variable, which is done via the bilateral conversation he or she has with the Bot. Or he or she can choose to continue to the next phase by typing *@group_model_building ready*, which tells the bot to go to the next phase.

4.2.3 Generate a detailed view of the problem variable

As this part of Group Model Building represents the Nominal Group Technique (see Section 2.1.2), we need to separate the participants from talking to each other or be distracted by each other. Therefore the bot will make clear a Nominal Group Technique session is about to start, and all the participants are invited to a individual bilateral conversation with the bot. In these bilateral conversations the bot will make clear that the users now can post all other variables that come to mind which have got something to do with the central variable. After they are ready, they can type *@group_model_building ready*. When they all are finished, the bot will post all the variables in *#masterthesis-demo*. All variables are posted as separate messages containing a couple of buttons:

- **Make a suggestion** If a participant wants to suggest an adjustment of this variable, he or she can push this button which will initiate a bilateral with the bot. After that, the bot will post in a thread to this variable that *user x has suggested y*.
- **I disagree** If a participant totally disagrees with the statement that this variable has something to do with the central variable, he or she can make that clear by pushing this button. The bot will ask hem or her why and after that has been answered, posts this as a thread message of the variable.³
- **I agree** If the participant agrees this variable to be here, he can push this button. The Bot will post this as a thread message.
- **I have a question** If the participant does not understand what this variable means, or does not understand why this is a connection, he or she can push the button. The Bot will ask the participant what his question is and posts this as a thread message to that variable.

As with the last step, the facilitator can modify the variables or delete them. If in the end all the participants agree, he can write *@group_model_building ready* which will notify the bot that it can go to the next phase. But before it does, the bot posts the results of the Nominal Group Technique session.

³In retrospect, this button is somewhat misplaced. In regular Group Model Building, the Nominal Group Technique phase is divergent, meaning that no variable is wrong. Later at the "finding causal relations phase", variables are agreed or not. In the testing that was done in this research (see Chapter 5 and Chapter 7), the button was only used once. It triggered the creator of the variable to explain the variable, because the disagreement was triggered by misunderstanding. Therefore, in the end the button only functioned in the same way the "I have a question" button did. One can also state that therefore the "I agree" button is also obsolete, and should be changed to "I understand".

4.2.4 Create causal relationships

As this whole implementation represents a sequential systematic approach of Group Model Building, the create causal relationships phase goes as follows. The Bot first posts which variable is central and then posts all other possible variables. The other variables are posted with the following buttons: +, -, +/-, and ?. The bot now asks the participants for every variable to vote on if they are a cause and with what relationship that is. So if the variable at question is *profit*, and the other variables on which can be voted are *revenue*, *cost* and *peanut butter* the participants will probably vote for *revenue* (+) and *cost* (-). Also here, for every variable a thread can be started if something is unclear about that variable.

If all is voted and discussed, the leading user now can make the votes final by selecting the button of the final relation type. A leading user can also chose to select noting, which means that variable is not seen as a cause at all.⁴ After a selection is made, the leading user can write *@group_model_building ready* to make clear the bot may continue. The bot does so by selecting one of the other variables to be the central variable now. All other variables are now put to question, please remark the original central variable is now also part of this list. The bot again asks the participants to vote for causes and the whole process continues. The bot will do this until all variables have been the central variable once. Next, the bot continues this whole cycle but now asks the participants to vote for effects of the central variable. This continues until all the variables have again been the central variable once. If this is all finished, the bot automatically continues to the next phase of Group Model Building.

4.2.5 Calculate and define end result

Given (as we have mentioned earlier) that our implementation will be a purely qualitative approach of Group Model Building, this phase consists of merely the posting of the results of the last phase, which looks like this:

```
profit:
  has as cause:
    revenue (+)
    cost (-)
  has as effect:
    peanut butter (+)
cost:
  has as cause:
    peanut butter (+)
  has as effect:
    profit (-)
```

After this has been produced, it is up to the leading user to find feedback loops. The bot can only explain what a feedback loop is, but it is up to the group to now discuss this in detail. Given that the causal relations already have been created, the risk of politics and bias are negligible (Vennix, 1996) (Rouwette and Franco, 2014).

⁴The voting system is only used as an indication for the facilitator (leading user) of a session. Voting is never definite, but can be a simple aid to see if were everybody stands. For instance, if all the participants vote +, it would be clear that discussion about that relationship is not necessary. But, if only half of the participants vote + and the other half something else, it is clear for the facilitator to start a discussion about it.

This concludes the implementation of Ouborg (2015). The implementation in Slack took about 3 months (not including the 2 months to understand the framework prior to this process). But it resulted in a tool that is usable in a distributed setting. In order to be sure it is a viable implementation for a Distributed Group Model Building session, next section will describe the first test session that was done using this tool.

Chapter 5

First impression in practice

The sequential approach mainly inspired by Ouborg (2015) had to be tested in practice in order to complete the Relevance Cycle of the Design Science methodology (Hevner, 2007). Therefore this chapter explains how a practical test was designed and what its basic results were. These results and feedback were used in combination with the theory of the Combined Approach to generate the final implementation of this thesis (see Chapter 6).

5.1 Testing the sequential implementation

The first ever test of our implementation as described in last section, was conducted at the Radboud University, Nijmegen. Five students were invited to be the test subjects, of which four were bachelor students and one was a master student. They had different backgrounds, two had a business administration background, one had a data science background and two had a computing science background. Four students had prior experience with Group Model Building (in the regular analogue style). The different backgrounds of all the students was on purpose. Their opinion on the first implementation could give a broader view of how the tool would work in different contexts. Also to have one student who did not have any experience in Group Model Building was on purpose, to see if someone could use this tool without any prior knowledge.

The testing setting was a small conference room. Every student had their own laptop. The writer of this thesis would act as the facilitator of the Group Model Building session. Everyone was not separated from each other and if they wanted, they could talk to each other, but they were instructed to communicate as much as possible via Slack. One hour was reserved for this test session, and the whole session was recorded on a video camera. The students were instructed to think aloud if something was unclear or if the tool reacted in an unexpected way.

After the test the students were given a questionnaire, which had the following questions:

- *Did you have (prior to the session) a basic understanding of the Group Model Building methodology? (yes/no)* This would make clear if the distributed tool was perceived differently due to their prior understanding of Group Model Building.

- *How many regular team collaboration sessions are you conducting every week? (>5, 3-5, 1-2, 0)* This would give insight in how experienced they were in team collaboration on a creative level.
- *What is your opinion on Group Model Building in general?* If a student hated the tool, this could rule out if this was because they dislike Group Model Building in general.
- *Do you think the use of a Slack Bot is an improvement on Group Model Building?* Can identify the viability of the distributed tool.
- *Did there occur any frustration s during the session when using the Slack Bot GMB? If so, please give examples why.* Is used to determine the potential pitfalls of the distributed tool.
- *What are in your opinion the pitfalls of using a Slack Bot in a Group Model Building session?* This could identify if the Slack platform or the tool causes the problems the participants perceive.
- *Could you suggest improvements for the Slack Bot?* Improvement suggestions can be used to improve this implementation by again following the Design Cycle.
- *Are there any other remarks you want to make?* Although these questions cover the largest part, there can always ideas or other problems a participant wants to express.

After the test session, the footage of the camera and the questionnaires were analyzed. It is not our intent to perform an empirical study on this test session, the analyzing is only to give a general impression on how the tool performs and not to fully one hundred percent prove it works well. As this is a proof of concept, a general impression would seem sufficient. s

5.2 General reception

The test was conducted as described above. There were no problems with Slack and the implementation never crashed. The students seemed to have no difficulties understanding Slack. During the testing, it became clear that the tool did not provide enough information about how to use it. All the students reacted somewhat promptly on when it was required to push the buttons and when to type in the chat itself.

Furthermore, it became clear that the creating causal relations phase was very badly implemented. Quickly students became very frustrated of the sequential flow of that process. Their suggestion was to make it an agile process, i.e. let the discussion determine which variable should be central at some point.

What went really well was the implementation of the Nominal Group Technique. Not a single student found it necessary to consult another student during this process, which meant it all happened silently. The process of Nominal Group Technique was followed flawlessly and resulted in a decent list of variables.

The results of the questionnaire were as follows:

- **Did you have (prior to the session) a basic understanding of the Group Model Building methodology? (yes/no)**

As was expected, four had experience with Group Model Building, one hadn't.

- **How many regular team collaboration sessions are you conducting every week? (>5, 3-5, 1-2, 0)**

Three students had 0 collaboration activities and two had 1-2 on average. This was somewhat surprising given the amount of collaboration tasks students have these days. This is possibly due to a misunderstanding of "team collaboration".

- **What is your opinion on Group Model Building in general?**

All the students claimed Group Model Building in general to be useful, but had side comments. Two students found that it took too long to complete and one student was afraid it loses its usability because "it can be hard to introduce" it to the general public. The latter was said by the student who had no prior experience with Group Model Building, therefore one can say that it is still a big step to use when you have no experience with Group Model Building yet.

- **Do you think the use of a Slack Bot is an improvement on Group Model Building?**

All students saw the Group Model Building Bot as an improvement, but for different reasons. One student said it was easier to learn because required actions were explained on the fly. One student said that the tool would be handy when a regular session was not possible, but still preferred the "face-to-face" form of Group Model Building. One other student said the fact that documentation is more easy is a great benefit. Given that it is integrated in your main chat application was seen as a benefit also. Last comment was that the Slack Bot, if worked perfectly, could be a great neutral mediator.

- **Did there occur any frustrations during the session when using the Slack Bot GMB? If so, please give examples why.**

There were a couple of frustrations felt by the students during the test:

- The Slack Bot contained typos.
- Messages were not always received in the right order as they were sent (Technical issue with the dedicated server).
- It was not always clear where to type.
- The overview was sometimes gone: The addition of drawings was suggested.
- Too much information at one, and too spread out, therefore it was very hard to get an overview of the variables and all the effects.
- Quickly lost track of comments and questions on variables.

This was coherent with what was seen in the video recordings and their think aloud commentary. Most frustrations were part of the failure of the causal creation phase.

- **What are in your opinion the pitfalls of using a Slack Bot in a Group Model Building session?**

The students had named a couple of pitfalls of the use of Slack for a Group Model Building session:

- Not being able to have quick discussion and see each others face was seen as a pitfall.
 - Expressing using text can be too hard, but this will work when used in the same room.
 - The fact that the Slack Bot lacks the feel of human interaction, misunderstandings can arise.
 - Lack of overview.
 - It takes too long, too much formal overhead, therefore difficult to get an agreement.
- **Could you suggest improvements for the Slack Bot?**

A couple of improvements were suggested:

- Adding the ability to go to previous sessions.
- Provide more information about how Group Model Building is structured during the process with the bot.
- Try to make the information more compact.
- More small steps instead of one large one.
- Should be possible to delete change and add variables during the causal relation phase.

This concluded the feedback of the students. As was expected, the need for a more agile approach was made clear. Therefore we can now state that the implementation of Ouborg (2015) is insufficient for a Distributed Group Model Building approach. This will be improved in the implementation of the combined approach which is explained in the next chapter. Given the feedback of the students, the following is taken into account when the improved version is being built:

- Redesign the create causal relations phase by making it more agile and less information intensive.
- Explain more what the structure of the process is.
- Explain where information can be found and what information is expected from a participant (and were to type that).

The next chapter will talk about the final implementation of this thesis. This implementation follows the combined approach including the feedback given in this chapter. Also, the next chapter will explain the works of the tool with the addition of explanatory figures.

Chapter 6

The final implementation

This chapter describes the final implementation of our Distributed Group Model Building implementation using the *combined* approach. It is based on a combination of literature by Ouborg (2015), Hoppenbrouwers and Rouwette (2012), literature of this thesis (see Section 2), and the feedback acquired from the first test (see Section 5). The chapter is built as follows: Section 6.1 shows how the final product looks like by explaining its steps which are accompanied by explanatory images. Section 6.2 describes the considerations that have been taken during the implementation of this approach, like which things have left out or are different than intended. This implementation forms the basis for our final tests that are described in the next chapter.

6.1 Implementing the combined approach

This section explains what the final implementation looks like. The explanation is accompanied by screen captures of the actual product and is mostly viewed from the perspective of the facilitator (called a leading user in Slack). This is due to the fact that the facilitator has the most possible operations he or she can execute, which makes it more interesting to show for how the tool works. In some cases the view of the participant (regular user) is shown, because some bilateral conversation between a participant and the Slack Bot is worth showing as the bot structures the opinion of the participant into a formal format. For readability and structure, the explanation is separated for every sub-step of Group Model Building.

6.1.1 Open and define rules

Our test takes place in a channel called *#thesisdemo*. Our leading user will be *@mata37* and there is a participant called *@mouborg*. Figure 6.1 shows how the starting situation looks like. Most of the time, a channel is already used for a purpose and will contain chat messages already. But for sake of this thesis, we have created a new channel. If we look at the starting situation, we see a banner on the left side which contain all the channels and users present in this Slack group. *#thesisdemo* is active at the moment and *@mata37* and *@mouborg* are online.

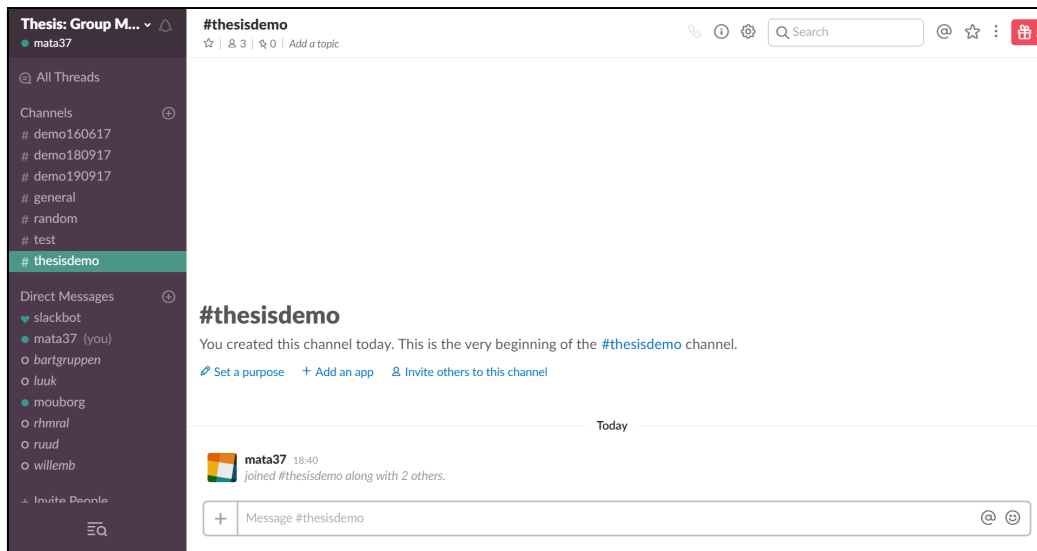


FIGURE 6.1: The initial state of the Slack application.

To start the Group Model Building bot, one has to type `/gmb` in the chat text field, as is shown in Figure 6.2. As one can see, there also exists a slash command to abort a Group Model Building session and one to get help.

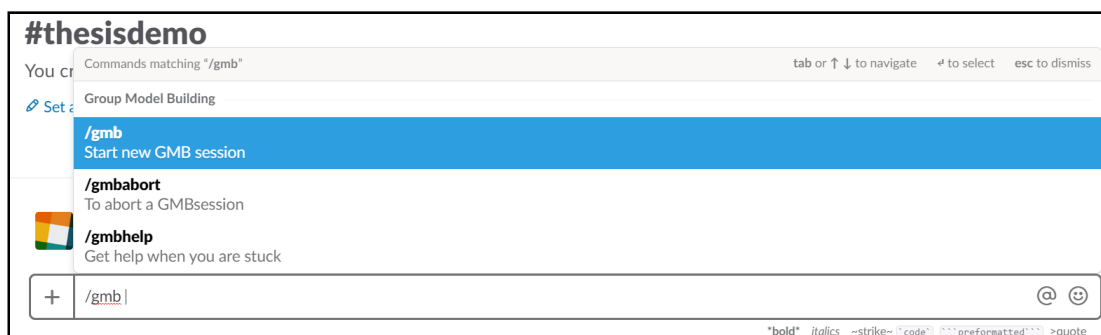


FIGURE 6.2: Starting of the Group Model Building bot.

If the bot has been called, it welcomes you to a Group Model Building session, as is shown in Figure 6.3. It also asks who the leading user will be. Remember that the leading user will act as a facilitator besides the Group Model Building Bot. The user in Slack who is supposed to be the leading user can now press the button "Leading user".

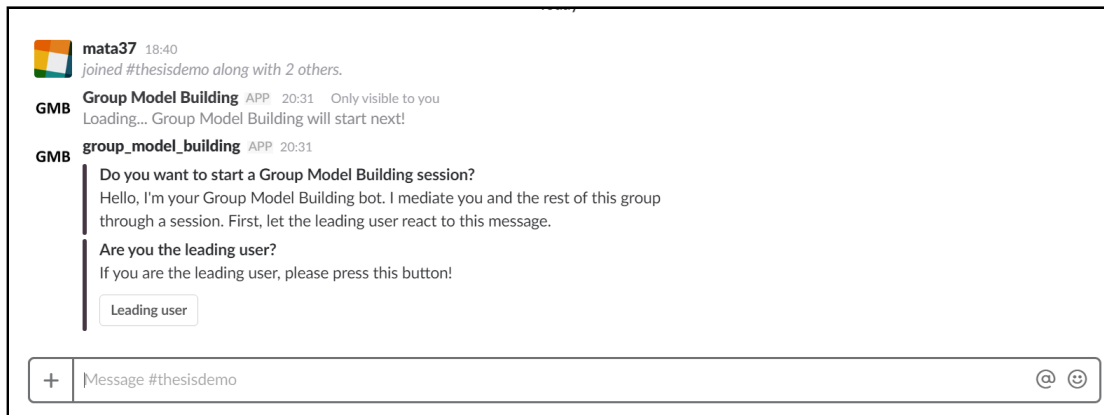


FIGURE 6.3: Group Model Building welcome message.

After the leading user pressed the button "Leading user", the bot continues to the next phase.

6.1.2 State the observed problem

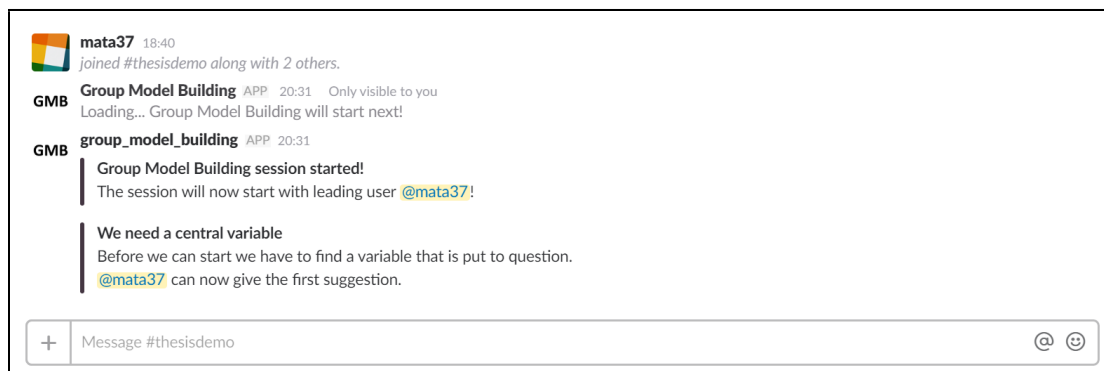


FIGURE 6.4: Group Model Building session has started.

As is shown in Figure 6.4, the Group Model Building Bot has now started the session. It makes clear @mata37 is the leading user and that he requested @mata37 for a main variable (variable of interest). For our leading user @mata37, the left border now looks like figure 6.5.

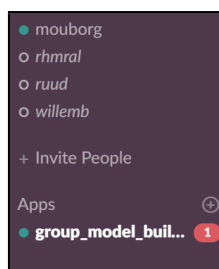


FIGURE 6.5: New message from the Group Model Building bot.

As one can see in Figure 6.5, the Group Model Building Bot lights up with a "one" after its name, which means there is one new message for @mata37 from the Group Model Building Bot. If @mata37 clicks on the Group Model Building Bot, as is shown in Figure 6.6, the personal question for a central variable is there.

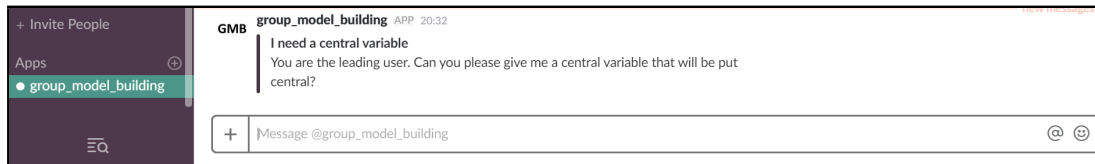


FIGURE 6.6: First bilateral message from the Group Model Building bot.

@mata37 can now simply give a central variable, one that has already been decided. Otherwise @mata37 can just give a variable to call into question. In this example @mata37 gives *profit* as a main variable, as is shown in Figure 6.7.

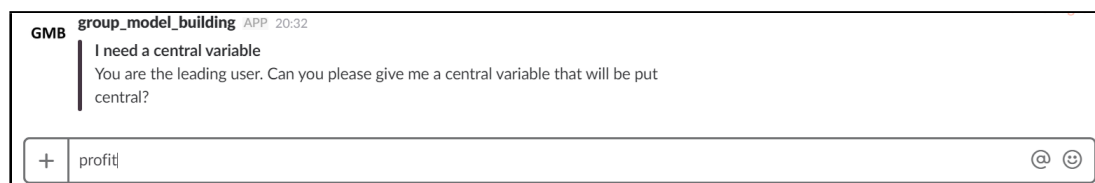


FIGURE 6.7: Suggesting profit as the main variable.

When @mata37 has entered "profit" as the suggested main variable, the Group Model Building Bot answers that the variable has been received, as seen in Figure 6.8. The Bot also states that if the leading user wants to change the main variable, he or she can write "@group_model_building change". Furthermore, it informs the user that the conversation now continues in the main channel #thesisdemo.

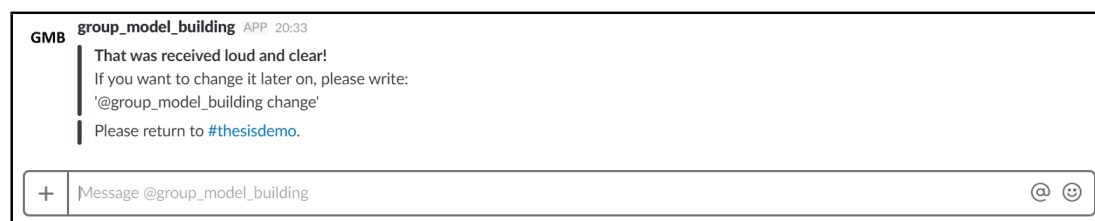


FIGURE 6.8: Main variable received and change instructions.

Now in the #thesisdemo channel, the Bot posted a message that the variable "profit" has been suggested by @mata37 (see Figure 6.9). This message is accompanied by four buttons. These are the same buttons as seen in Chapter 4.

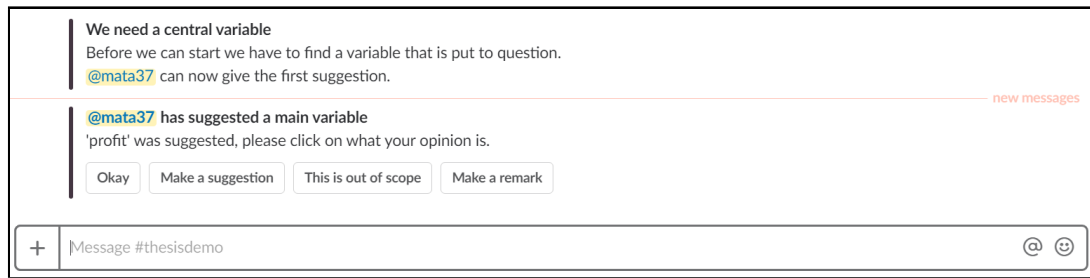


FIGURE 6.9: Main variable message in main channel.

The buttons represent the same possible actions:

- **Okay** With this button, participants can easily make clear that they agree with the suggested central variable. This will create a thread message that states that *x agrees!* (with *x* the user who pushed the button).
- **Make a suggestion** This button can be chosen if a participant wants to suggest another variable or an adjusted one. This will trigger a bilateral between the that participant and the Group Model Building bot in which the bot asks questions so that the opinion of the user is according to a predefined format. After the questions have been answered, the bot posts a message to the thread of the central variable and states that *user x suggested y because z.*
- **This is out of scope** Just like *Make a suggestion* the bot will start a bilateral conversation with the participant who pushed this button and will ask the participant why he or she thinks the current central variable is out of scope. The bot then posts in the thread of the central variable that *user x thinks this variable is out of scope, because y.*
- **Make a remark** If a participant just has something to say about this variable, he or she can push this button. This will also trigger a bilateral conversation between the bot and the participant, and the bot will ask what their remark is. After this has been answered, the bot will post a message in the thread of the central variable, stating that *user x has a remark, namely: y.*

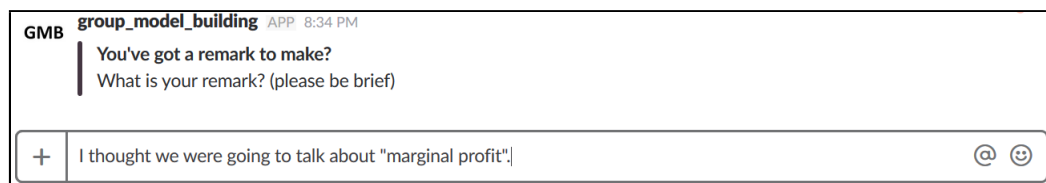


FIGURE 6.10: Remark of participant in bilateral conversation with the bot.

In our example, participant @mouborg has a remark to make and presses the "Make a remark" button. This results in a bilateral conversation with the Group Model Building bot, as is shown in Figure 6.10. The bot asks a question and @mouborg answers.

When @mouborg has submitted his answer, the Bot make clear to @mouborg the conversation continues in #thesisdemo. What has changed to the original post of the main variable, is that the Bot has posted a reply to the message (as is seen below the buttons in Figure 6.11).



FIGURE 6.11: A reply to the original message.

If we open click on the reply notification, Slack automatically opens the corresponding thread (see Figure 6.12).

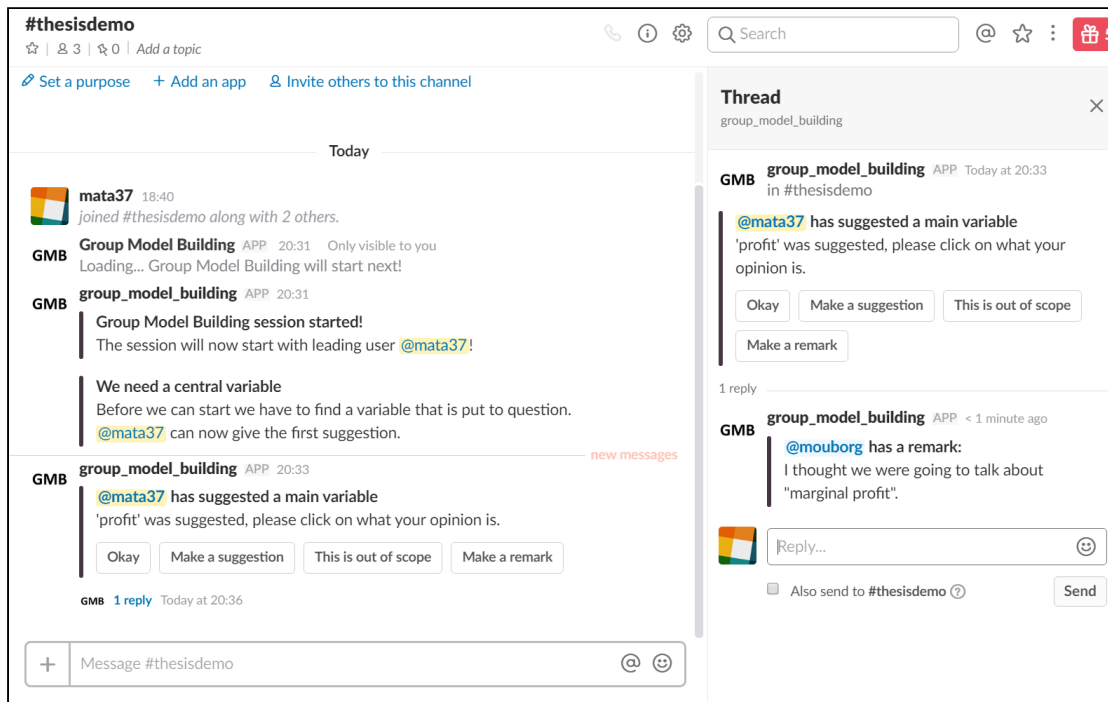


FIGURE 6.12: A thread has been opened by Slack.

If we look closer at the thread, we can see that the Group Model Building Bot has posted a message that @mouborg has a remark and information about this remark. The reason we let the Bot post the messages of a remark and not the participant itself, is because we want all the reactions in a strict and same format. This prevents the answers from being misunderstood. For instance, if all the answers are of the format ”@mouborg has a suggestion: -suggestion-”, it is immediately clear the reaction is about a suggestion and what that suggestion is.

Given that the response is of an unemotional nature and posted by the neutral Group Model Building Bot, participants can now react to this message if they want. Figure 6.13, shows the reaction by the leading user @mata37 and the reaction of @mouborg.

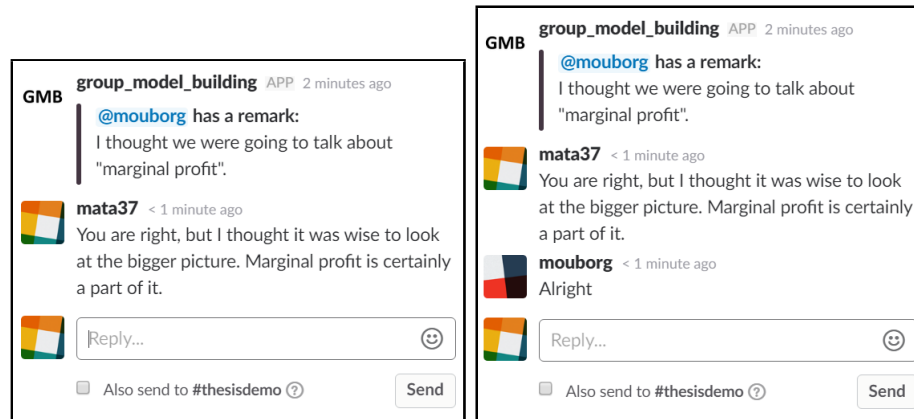


FIGURE 6.13: Leading user reacts to remark.

If there is consensus about the main variable, the leading user can write ”@group_model_ouborg ready” (see Figure 6.14) in the main channel to let the Bot know it can continue with the next step of Group Model Building.



FIGURE 6.14: Leading user lets bot know it can continue.

This will force the Bot to continue with the next step in Group Model Building, namely *Generate a detailed view of the problem* which is explained in the next section.

6.1.3 Generate a detailed view of the problem

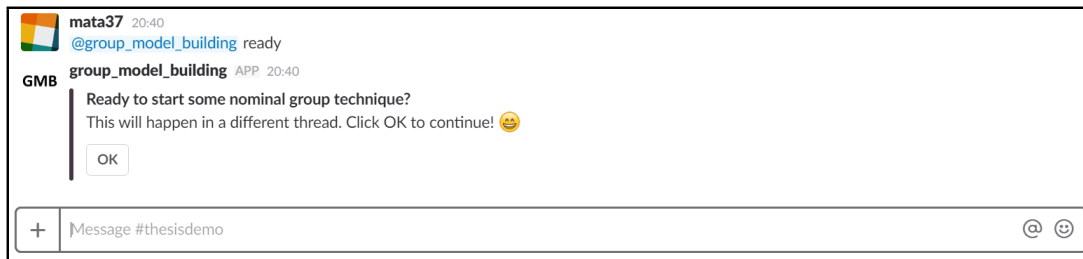


FIGURE 6.15: The bot wants to start a Nominal Group Technique session.

If the leading user has made clear the Bot can continue, it will post a message that a Nominal Group Technique session is about to start (see Figure 6.15). The "OK" button is there for technical reasons. Anyone is allowed to press it and if one presses the button, the session will start for all, as seen in Figure 6.16.



FIGURE 6.16: Nominal Group Technique started in separate threads.

If a Nominal Group Technique session is started, the Bot will create a bilateral conversation with all the active participants. It posts a message that one can now write down variables that come to mind when thinking about the central variable "profit" (see figure 6.17).

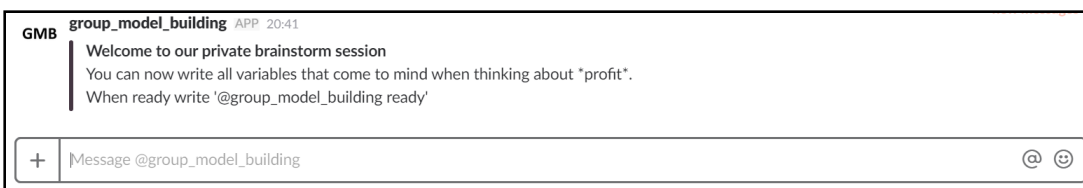


FIGURE 6.17: Bilateral conversation for creating detail.

Figure 6.18 is an example of variables that have been created by a participant in this bilateral conversation.

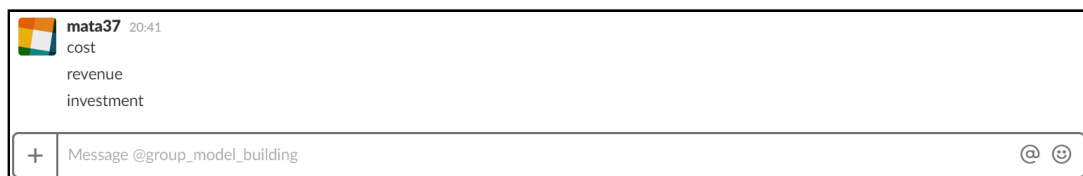


FIGURE 6.18: Created detail in the bilateral conversation.

If the participant is sure he or she has no more variables to add, he or she can write ”@group_model_building ready” in the bilateral chat to notify the Bot he or she is ready (see Figure 6.19).

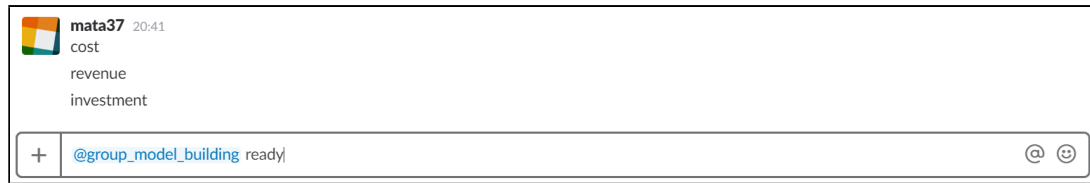


FIGURE 6.19: Letting the bot know a participant is ready.

If the participant has posted his ready message, the Bot will answer with a question if the participant is sure he or she is ready. This is to make sure the command did not go on purpose (see Figure 6.20).

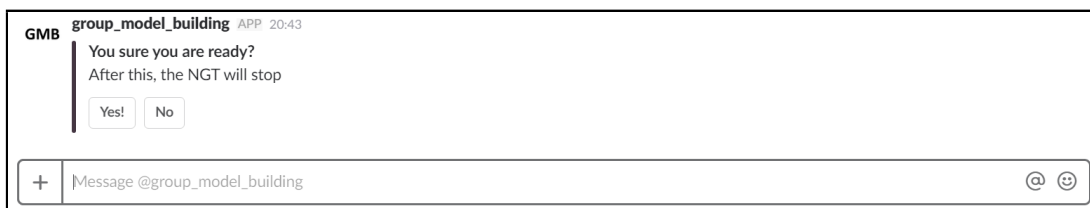


FIGURE 6.20: Making sure the participant is ready.

Figure 6.21 shows that the Bot has received the information that the participant is ready, but it also lets the participant know that others have not finished yet.

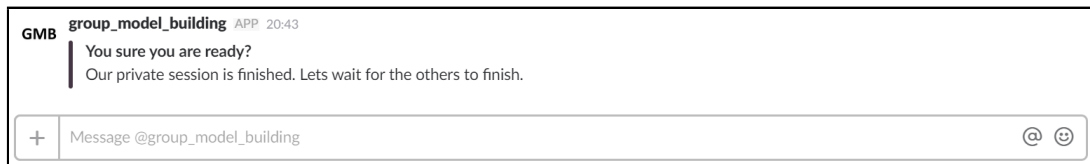


FIGURE 6.21: Waiting for other participants to finish.

If the other participants have also finished their private Nominal Group Technique session, the Bot lets everyone know the session is finished and that everyone should return to #thesisdemo (see Figure 6.22).



FIGURE 6.22: Nominal Group Technique session is finished.

When all the participants return to the main channel #thesisdemo, they will see an overview of the generated variables (Figure 6.23). The Bot posts every variable that has been produced by all the participants (duplicates are filtered). For every variable, the participants can say what they think about that variable by pushing on one of the four buttons.

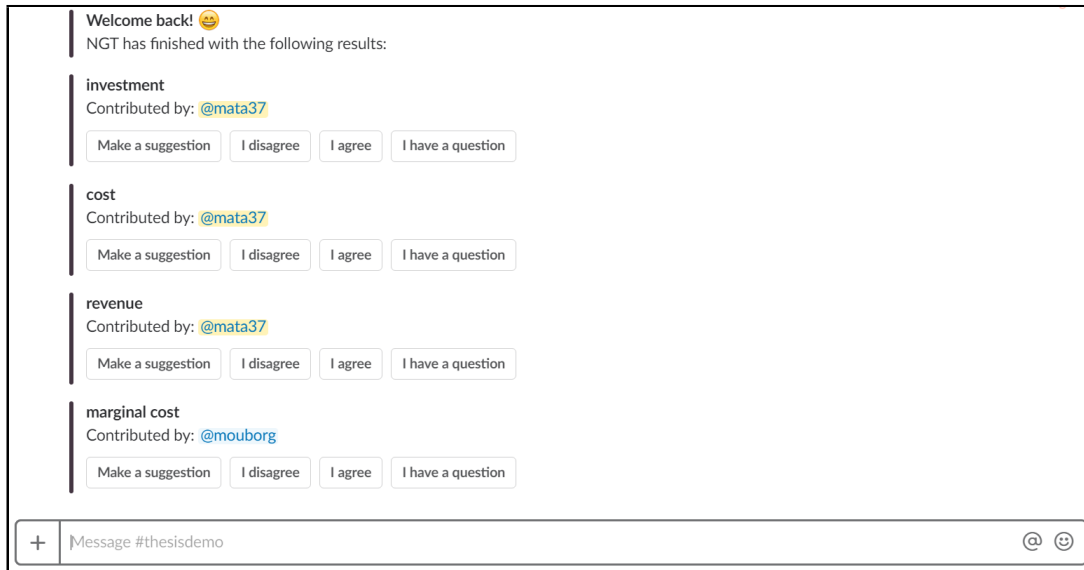


FIGURE 6.23: Results of the Nominal Group Technique.

The buttons are the same as described in section 4.2.3 and mean the following:

- **Make a suggestion** If a participant wants to suggest an adjustment of this variable, he or she can push this button which will initiate a bilateral with the bot. After that, the bot will post in a thread to this variable that *user x has suggested y*.
- **I disagree** If a participant totally disagrees this variable has something to do with the central variable, he or she can make that clear by pushing this button. The bot will ask hem or her why and after that has been answered, posts this as a thread message of the variable.¹
- **I agree** If the participant agrees this variable to be here, he can push this button. The bot will posts this as a thread message.
- **I have a question** If the participant does not understand what this variable means, or does not understand why this is a connection, he or she can push the button. The bot will ask the participant what his question is and posts this as a thread message to that variable.

¹(As was also mentioned in Chapter 4:) In retrospect, this button is somewhat misplaced. In regular Group Model Building, the Nominal Group Technique phase is divergent, meaning that no variable is wrong. Later at the "finding causal relations phase", variables are agreed or not. In the testing that was done in this research (see Chapter 5 and Chapter 7), the button was only used once. It triggered the creator of the variable to explain the variable, because the disagreement was triggered by misunderstanding. Therefore, in the end the button only functioned in the same way the "I have a question" button did. One can also state that therefore the "I agree" button is also obsolete, and should be changed to "I understand".

In our example, @mata37 agrees with cost. He pushes the button and the bot immediately shows in a thread that @mata37 agrees (see Figure 6.24).

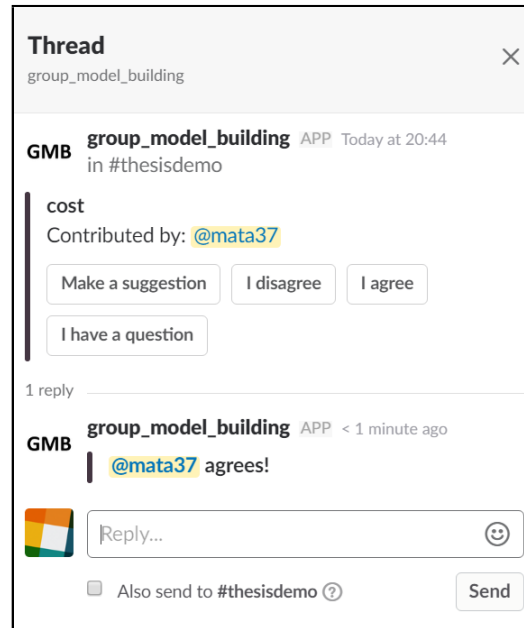


FIGURE 6.24: Participant agrees with variable.

In this phase, the leading user @mata37 has a lot of options to adjust the variables. For instance, when @mata37 as the facilitator wants to delete a suggested variable (because this was agreed in the group), he can write "@group_model_building delete" as a bilateral message to the Bot. The Bot will respond as seen in Figure 6.25, asking for the variable to delete.

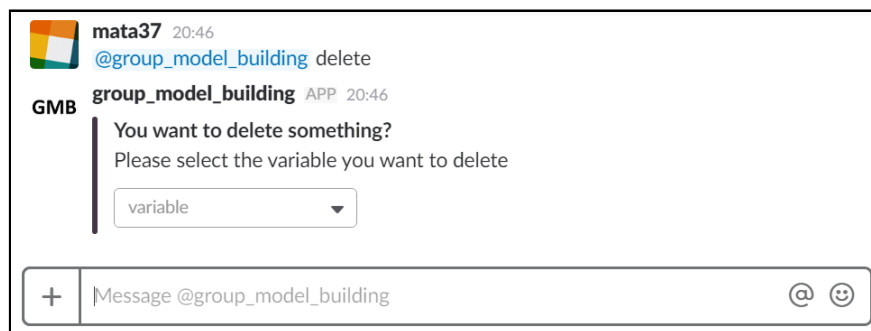


FIGURE 6.25: Deleting a suggested variable.

In our example, @mata37 chooses "marginal cost" as the variable that has to be deleted, given that the group agreed to do so. Figure 6.26 shows how this looks in the main channel #thisisdemo. As one can see, the variable "marginal cost" is still there, but it is made clear the variable is deleted by the leading user.

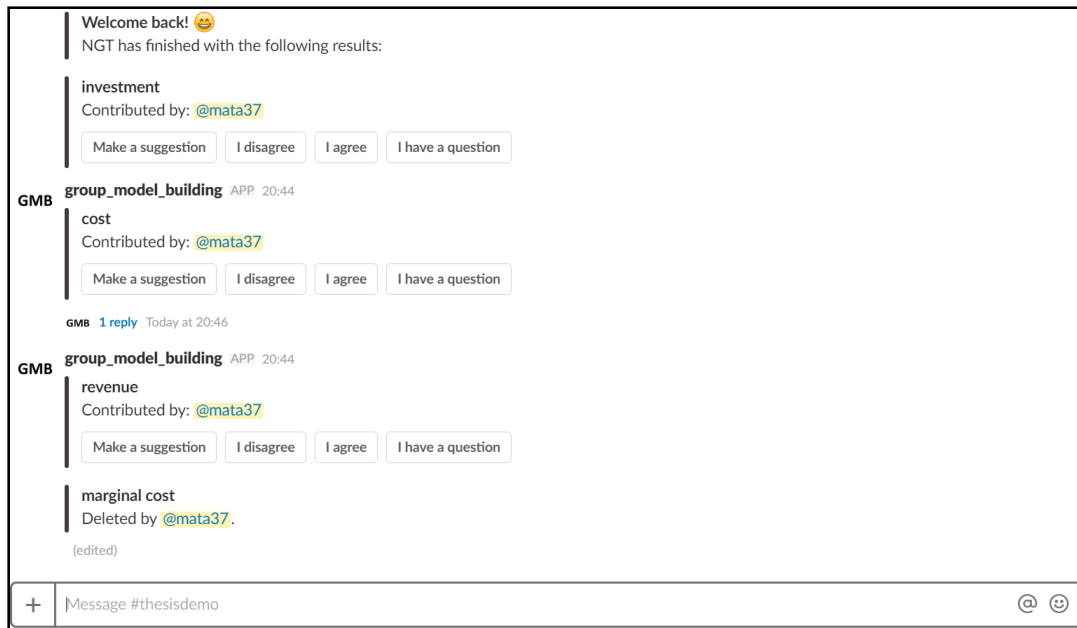


FIGURE 6.26: A variable has been deleted.

If all the adjustments have been made and the discussions have finished, @mata37 can make clear the group is ready to continue to the next phase of Group Model Building by typing "@group_model_building ready" in the main channel. If the Bot receives this, he will continue.

6.1.4 Create causal relationships

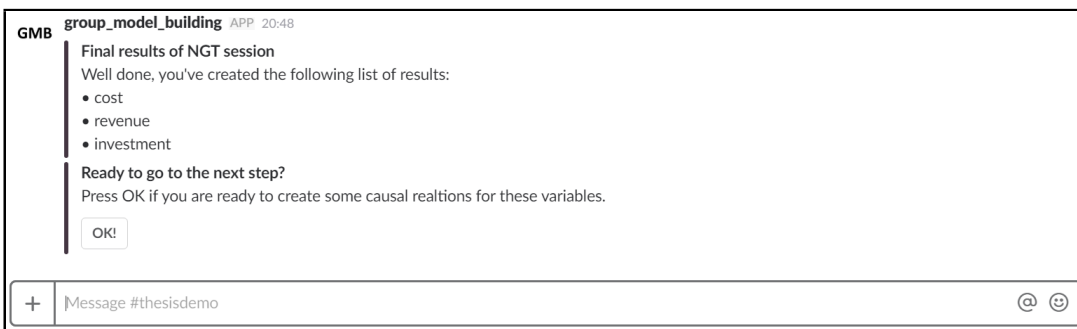


FIGURE 6.27: Nominal Group Technique phase has ended.

As shown in Figure 6.27, the bot has started the next phase in Group Model Building. First it gives an overview of the variables that have been created and asks the participants to press "OK". As was mentioned in the last chapter, this button is there due to technical reasons.

If one of the participants have clicked on "OK" the bot will let all the participants know their should be looked for causal relations. Figure 6.28 shows this message, which also states that @mata37 should initiate the discussion.

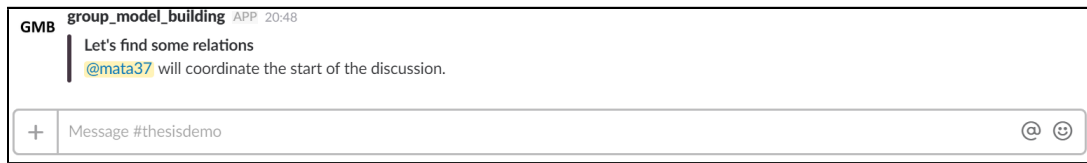


FIGURE 6.28: Start of the causal relations phase.

This approach is the result of a complete redesign of this phase. To make it agile, instead of sequentially walking through all the variables, the leading using now has a control panel in his bilateral conversation with the bot, as shown in Figure 6.29.

FIGURE 6.29: Control panel for the causal relations phase.

In this control panel, the leading user can select a variable that is put central for that time being. If a central variable is chosen, the leading user can select causes and effects with their corresponding relationships to the central variable. This happens of course after discussion takes place. Lets say the leading user selects "profit" as the variable of interest. In the main channel this is made clear as shown in Figure 6.30.

FIGURE 6.30: Variable of interest is chosen.

As one can see in this overview, the central variable is shown with a couple of buttons, also an overview of the other variables is given and the so far found connections are showed (in Figure 6.30 this is still empty).

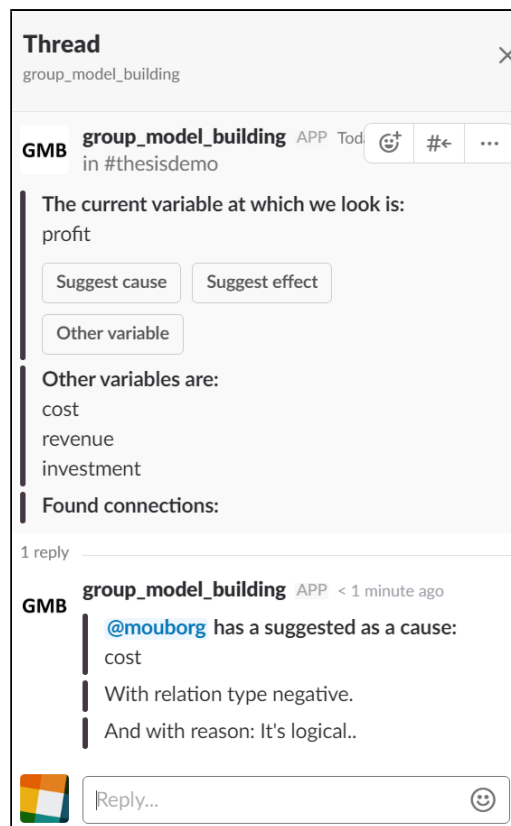


FIGURE 6.31: Thread for cause suggestion.

Figure 6.31 shows the result of what happens when a participant suggest a cause. The Bot contacts that participant after he or she pressed the button "Suggest cause" and asks a couple of questions (in order to formalize their response). In this case @mouborg has suggested cost as a negative cause to profit. If after discussion it is clear that the participants agrees, the leading user @mata37 can add the variable as a cause to profit. This process will continue and after a while will look as shown in Figure 6.32.

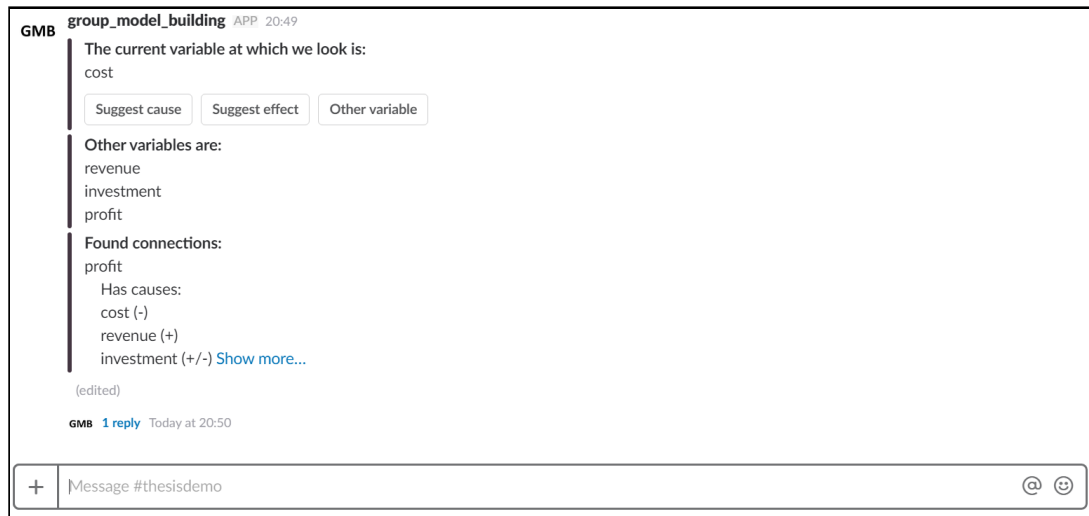


FIGURE 6.32: Overview after a discussion.

Here one can see that cost is now central at the moment, and there are a couple of causes found for profit. If the whole discussion has ended, it is upto the leading user as a facilitator to explain these found relations and search for possible feedback loops which happens in the next phase of Group Model Building: *Calculate and define end result*.

6.1.5 Calculate and define end result

In this last phase of Group Model Building, feedback loops are explained. In our approach this is totally up to the leading user to do. This happens by typing in the main channel what his thoughts are. Of course discussion can take place, but in this phase this is allowed. All creative ideas have already been posted and are not vulnerable for bias or politics.

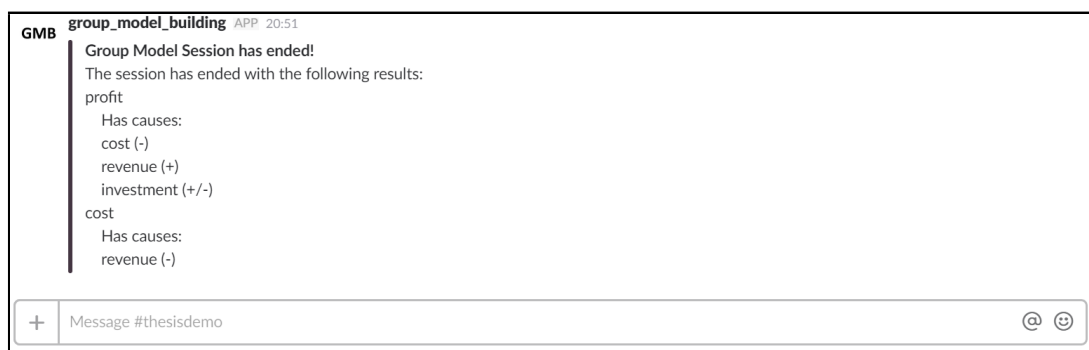


FIGURE 6.33: Overview after a the session has finished.

If in the end all is clear, the leading user @mata37 can type "@group_model_building ready" to let the Bot know they are ready. Given that this was the last part of the Group Model Building session, the Bot will give an overview of the causal relations that have been found (Figure 6.33).

Next section describes the considerations that have been made during the building of this implementation. The building itself (given that it is an adjustment of the first sequential implementation), took about 3 months. A lot of technical difficulties arose, like concurrency issues. Given that the order in which messages are received should be correct (otherwise misunderstandings will take place), these error had to be corrected and could not be pushed under the name of "proof-of-concept".

6.2 Considerations

During the development of the implementation described in last section, a couple of considerations were made. During development it became clear that some aspects were highly complicated to digitize using the BotKit (see Chapter 4) and Slack. Therefore some parts have been left out. For instance, in an ideal Group Model Building situation when you are in the causal creation phase, it can occur that you want to add an extra variable to the list of variables. This can be logical in the case a variable's relation does not seemed definable and the group senses some information is still missing. A connection $A \Rightarrow B$ can then be extended to $A \Rightarrow C \Rightarrow B$ (with C as the new variable) which makes it more logical. In our approach this extra variable could not be added during the causal relation phase, as this was too complex to program within the time scope of this project.

Also, the implementation of live modeling was left out. Therefore linking variables was only shown as a textual list of variables with their causes and effects and not as a visual causal loop diagram. In regular Group Model Building, the collaborative building of the diagram creates the understanding and consensus one is looking for. It is the foundation of the statement that "Structure drives behavior" (Vennix, 1996) (Rouwette and Franco, 2014). This is therefore seen as the biggest pitfall of the proof-of-concept, created in this thesis.

Furthermore, as was already mentioned, the suggestion of feedback loops is still up to the facilitator as if it were a normal chat application. The Bot has no influence here. With a more advanced Bot, the Bot could suggest feedback loops by itself and leaving it upto the group to select the logical ones. This would not only have sped up the process, but also prevent the chat from being too chaotic and unclear. Last consideration is that of the causal loop diagrams that are missing in this implementation. The reason for this is the same as of the ones above: It simply was too complex to program within the time scope. In the future, an advanced Bot can be implemented with automated drawings but this is for now a long way off.

In short, the considerations that have been made (due to technical feasibility) are as follows:

- Adding extra (new) variables to the list of variables during the causal creation phase was not possible.
- Live modeling was not implemented and created relations where shown in a textual way.
- Automatic feedback loop suggestions were not implemented.

Nevertheless and all considerations into account, it does not mean the tool is weak. The Nominal Group Technique is nearly a perfect digitized representation of the original analogue version. Also, in the end, variables can be linked to each other while bias and politics was prevented at the necessary phases. It is true that one of the biggest strengths of Group Model Building is the live modeling which give visual insight in the relations, but as said before this tool can be extended with that feature in the future. Until then, this implementation should be viewed as a pioneering proof-of-concept which shows that Group Model Building can be easy to use when it is implemented in a regular chat with the support of a Bot.

To complete the Design Cycle, this tool has to be tested in practice. The following chapter explains how two tests were conducted. One with the same group of student, another with experts in the field of Group Model Building. Both tests will determine if the implementation described in this chapter is a useful implementation of Distributed Group Model Building.

Chapter 7

Testing the final implementation

This section describes the last two tests that were conducted. In these tests, the final implementation as described in last chapter was tested. One test was conducted with almost the same group of students as the last test, the other test was conducted with three experts in Group Model Building. These last test hopefully give insight in the viability of the approach chosen in this thesis.

7.1 Designing the last tests

As was mentioned before, two test were conducted. The tests each had a different goal in mind. The test with the students was more in the form of seeing if the final implementation was seen as an improvement and the test with the experts was more a test to see if the approach in general was seen as viable. We first describe how the test with the students was conducted and then how the test with the experts was conducted. Next section describes the results of these tests.

The test with the students was conducted with almost the same group of students as last test. Only the student with the data science background was prevented from joining. For the research this had no consequences given that there still were some students with more or less the same skills set and background. So the test was conducted with four students. As a subject for the Group Model Building session was chosen the same subject as with the last test. This was chosen so the focus was to look if they found the new implementation more usable.

The test took an hour and the students were again asked to think aloud when something was unclear or if a frustration occurred. Also this session was recorded on video, and later analyzed for the results (see section below). It was made clear to the students that they also had to think about the question if this implementation was an improvement over the last one. At the end, they were given the same questionnaire (see Section 5) with the addition of a compare question.

The second test was conducted with the help of three Group Model Building experts. They are all experienced Group Model Building users and facilitators. Also, they teach in the course Intervention Methodology and are thus aware of the Group Model Building

context and scientific background. Compared to the group of students, they had less experience in online tools such as Slack. This was taken into account during the test.

The test itself was of an informal nature. Its general goal was not to prove if the benefits of Group Model Building worked by simulating a Messy Problem. Instead, the goal was to walk through the processes of the Bot while relying on the experience of the experts. With this is meant that they were asked to identify any potential problems that the tool created. In other words, if the tool did not support all the benefits of Group Model Building, the experts were asked to mention them.

This test took about an hour and began with an extended explanation of Slack in general, given that the experts were not familiar with the platform. Also given that their background was not assumed technical, support for Slack on the fly was expected necessary. After the explanation of Slack, a simple demonstration of the tool was given by showing all its functionality. The experts were asked about how they saw the usability of this functionality in practice. After the demonstration was completed, the experts were given a questionnaire. This was the same questionnaire as the students got, without the compare question (see Chapter 5).

After both sessions were completed, the questionnaires and footage were analyzed.

7.2 Results after testing

The test with the students

On a technical level, the test with the students did not go that well. A lot of times the internet connection of one of the devices failed, and concurrency issues arose. This created an unworkable situation that resulted in a crash of the Group Model Building Bot. After a clean start, the Bot worked again and the test could continue. Although the implementation is nothing more than a proof-of-concept (which means technical issues are allowed), large technical issues could create frustrations which influences the opinion of the participants of the test.

Given that the students were already familiar with the Group Model Building Bot principle, the testing process itself went flawless. It looked like the students had learned how to use the tool, given that they reacted on comments a lot faster. Also compared to the last test they have conducted, fewer questions were asked were thing could be found. It is hard to tell if this was due to the extra information messages the Bot provided (like: "now we continue in a thread" etc.), or if this was due to the fact that they have simply learned how to use Slack and the Group Model Building Bot.

The questionnaires which were filled in by the students at the end of the test, gave insights to the following:

- **Did you have (prior to the session) a basic understanding of the Group Model Building methodology? (yes/no)**

Whereas in last test one student had filled in that he had no prior experience in Group Model Building, now all the students stated they have prior experience in Group Model Building. This is of course logical given that the former test can be seen as prior experience.

- **How many regular team collaboration sessions are you conducting every week? (>5, 3-5, 1-2, 0)**

The number of team collaboration activities remain the same. By deduction it became clear the missing Data Science student was one of the 0 collaboration activities on average.

- **What is your opinion on Group Model Building in general?**

Their opinion of regular Group Model Building generally remained the same. Only two new difficulties of regular Group Model Building were mentioned. *It requires prior knowledge of the method* and *It requires a highly skilled facilitator* were newly made remarks.

- **Do you think the use of a Slack Bot is an improvement on Group Model Building?**

The students were still positive about the addition of a Slack Bot, and this time used the following arguments:

- *"It is neutral in the sense that humans have an opinion about possible suggested ideas."*
- *"Yes, in the case of a multinational with different locations over the world, it can be a good solution. But only if all the details the technique recommends are thought of."*
- *"Yes, it creates a good overview and you can all work at the same time. You do not always have to wait for each other. (...) Sessions are saved and it is easy to review the results online."*
- *"I think it is, or at least could be, because it removes the absolute need for participants to be in the same room for the brainstorm session. The thing that has to be really kept in mind is that some of the nuances and explanation normally happening in a conversation (face-to-face), should now be done via the bot. This not only requires the bot to support this functionality, but more importantly: the participants need to use this technique, even if only one of them is not in the same room as the others."*

Their comments clearly show that they see the tool's potential, but also see its risks. Certainly the last comment shows these concerns. We will elaborate this more at the end of this section.

- **Did there occur any frustrations during the session when using the Slack Bot GMB? If so, please give examples why.**

Compared to the last test the following changes in frustrations were present:

- The technical issues in messaging remained, be they of a different kind. This resulted in a felt lack of feedback, which sometimes confused one of the participants. This was also still clear from the video footage.
- Slack is sometimes slow, which resulted in a misunderstanding about if a written message was received. The video footage confirmed this. Also hiccups of Slack were mentioned.
- Switching between channels (from a bilateral to a main channel and back) was seen as frustrating. In other words, constantly have to do something while this was felt as unnecessary.

- No remarks of typos were mentioned. Also not knowing were to type (missing information) was not mentioned anymore. Last, too much information at once was not mentioned.
 - The different threads were seen as confusing. Mainly keeping an overview of them and being updated if threads contained new messages was seen as confusing.
- **What are in your opinion the pitfalls of using a Slack Bot in a Group Model Building session?**

The potential pitfalls of the Group Model Building bot mentioned in the first test, remained the same. Only one new pitfall was mentioned: *"A computer display is smaller than a (white)board. So you might lose your overview during the session on a smaller display."*

- **Could you suggest improvements for the Slack Bot?**

Suggestions for improvement were slightly different from last test:

- *More small steps instead of one large one and Try to make information more compact* were not mentioned any more.
 - *Make notifications of Slack more clear* was new and added as a suggestion.
 - *Announce that a variable has changed* was new and added as a suggestion.
 - *Make face to face communication possible* was new and added as a suggestion.
- **Compared to the last test, what is your opinion of this implementation?**

The compare question resulted in the following remarks:

 - *"The latter way was better as the first time there was a lot of 'bloat' in the channel, which caused some confusion as to what you were discussing."*
 - *"(...) There is now a chance that not all variables are reviewed, because participants have to review them one at the time and are not forced to do so. There should guarded for a balance."*
 - *"The good thing compared to previous session was that it was much more compact. This made it clearer for us what we were doing. Furthermore the voting of the effect were gone. You could just fill this in yourself."*
 - *"This version is clearly an improvement over the earlier version, which was not agile. Especially for demo purposes, one would like to skip through variables, or give priority to them. Another improvement is in the usability, because Slack does not give you all the information you need to process at once."*

The opinion of the students clearly shows, that the final version of the Group Model Building Bot is an improvement over the first. But, big frustrations remain. Although the Bot is seen as a structuring help and prior knowledge of the steps of Group Model Building are not necessary anymore, the lack of human-like interactions remains a big missing element. Even if there is a facilitator who is part of the session in Slack, the need for face-to-face communication remains present. We will go into deeper detail in Chapter 8.

The test with the experts

During the test it became immediately clear that the lack of understanding in Slack was not resolved by a crash course of 15 minutes. During the whole test session a lack of feeling with the platform in general was sensed. Also the video footage clearly shows the frustration of the participants constantly present because participants could not find messages or did not understand the notification automatically generated by the Slack platform.

Therefore it was expected this influenced the final opinion of the experts. But, during testing it became clear they were understanding that the technical aspect was not the element that was main in this research. They were constantly giving constructive feedback and mentioned the potential pitfalls that could arise during the process. Together with the results of the questionnaires, the feedback was as follows:

- **Did you have (prior to the session) a basic understanding of the Group Model Building methodology? (yes/no)**

Logically they all have prior experience in Group Model Building.

- **How many regular team collaboration sessions are you conducting every week? (>5, 3-5, 1-2, 0)**

On the question how many times they use team collaboration techniques a week on average. Only one of the three experts said 1-2 and the others said 0. Given that the experts are constantly working on the context of Group Model Building, it was expected the question was not clear to them or was understood in another way.

- **What is your opinion on Group Model Building in general?**

Logically they are all positive about regular Group Model Building in general and gave the same arguments as the students did.

- **Do you think the use of a Slack Bot is an improvement on Group Model Building?**

On the question if the Group Model Building bot could be seen as an improvement over regular Group Model Building, their answers were the following:

- *“The improvement lays largely in the fact that participants do not have to join together. Furthermore all participants can give their opinions and this is registered. At a real session, the facilitation has to watch body language closely to see if everybody agrees. Some stakeholders are doubtful, but do not say a thing.”*
- *“Only for non-messy problems (and then you perhaps don’t need gmb). Psychological processes in gmb so important (different stakeholders with different interests, different values, different goals) but they cannot be addressed.”*
- *“No, you are missing the social side of GMB strongly, the real collaboration of a group. Especially if it is about sensitive subjects, face-to-face contact is really necessary. The test we conducted was of course short and mostly on the technical part, therefore it is difficult to judge what the benefits of the tool could be.”*

- **Did there occur any frustration s during the session when using the Slack Bot GMB? If so, please give examples why.**

The following frustrations arose:

- Slack works cluttered. In real Group Model Building sessions an enormous list of variables can be created, which in Slack results in an large cluttered list of variables, which in return downplays the benefit of Group Model Building.
 - Not clear what the direct response on own created variables are. The thread system in Slack does not seem to give enough heads up when necessary.
 - No sense of interaction with the participants.
- **What are in your opinion the pitfalls of using a Slack Bot in a Group Model Building session?**

According to the experts, the pitfalls are:

- Because of Slack: Sensitive subjects will be more difficult to expressed. (Even with the addition of Skype) it is difficult to create a feeling of being heard. You do not know them and you are not starting to know them either. A participant can easily miss things.
 - Strict problem-focus disabling forging more social connection between participants, therefore too little social contacts.
- **Could you suggest improvements for the Slack Bot?**

Their suggested improvements were the following:

- Create a better overview
- Rethink what information is necessary at some point¹.
- Make it technically more easy and try adding Skype.

It is clear the experts were not as convinced about the potential of a Group Model Building Bot as the students were. It is difficult to express what causes this difference in opinion. Possible factors are of course the difference in experienced in digital tools, but also expectation levels might differ. With the latter is meant that the students know Group Model Building at a very basic level, while the experts know lots of detail. In the mind of the expert a good Group Model Building process has to comply with a lot more requirements than the basic requirements described in this thesis. Therefore, it can be easier to fail the requirements of the experts. But, it must be said their feedback is certainly seen as important and of value, and is certainly not trivialized. Therefore next chapter (Chapter 8) will discuss the results of both test of this chapter, and philosophizes the role of a Group Model Building bot in the future of team collaboration.

¹This remark was made after the list of variables disappeared when trying to create causal relations. The disappearance was mostly due to technical difficulties with Slack.

Chapter 8

Results and discussion

Last chapters described the created implementations of a Distributed Group Model Building tool. Two implementations were made and three test were conducted. This resulted in three video recorded sessions and three sets of questionnaires. Chapters 5 and 7 showed the results of those tests by explaining what was remarked during the test and written in the questionnaire afterwards. This chapter combines all the results and links it to the original theory of Chapter 2. This will lead to a discussion which gives a hint in how future proof this implementation style of a Distributed Group Model Building tool is.

At the end of this chapter a conclusion to the second research question "*Does a Distributed Group Model Building tool using Slack enable the benefits of regular Group Model Building?*" is given. This question represented the *Evaluation question* which is part of the second step in the Design Cycle of Design Science (see Section 1.2). In order to fully formulate the result according to Design Science, the steps as described in Section 1.2 (Wieringa, 2009) are used.

8.1 Completing the design cycle

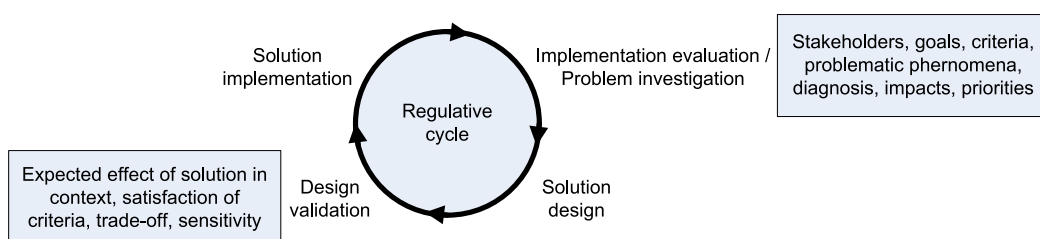


FIGURE 8.1: The regulative cycle of Design Science (Wieringa, 2009).

Before we can evaluate what the results of the conducted tests tells us, we have to elaborate this research according to the design cycle in order to prove its legitimacy. This will be done by the eight guidelines which clarifies the "*interface of design science with its social environment and with the scientific knowledge base*" (Wieringa, 2009). For purpose of readability, we will go step-by-step through these guidelines (Wieringa, 2009):

1. **Distinguish practical problems from knowledge questions**

This was done by first elaborating the necessity of a Group Model Building tool in general. Given the growing number of organizations operating at a global level, it becomes harder and harder to get every stakeholder in one room together. The need for a distributed approach in solving Messy Problems is therefore eminent. This describes the practical problem. The knowledge questions that automatically followed, were described in the Design Research question.

2. **Solve practical problems by the regulative cycle**

The regulative cycle as shown in Figure 8.1, consists of four steps. By first describing the necessities of a Group Model Building tool in Chapter 2 and then implementing two implementations of this theoretical research, we went through this Regulative cycle three times. Always starting with the "Problem investigation" and ending with the "Implementation evaluation" step. In the next section the final Implementation evaluation is conducted.

3. **Distinguish problem investigation from design validation**

This distinction is made by first describing what is necessary by using theory and then describe a implementation of the tool. The expected behavior of the tool is then described by connecting the suggested implementation to the theory. In the second and third cycle, the problem investigation consisted of the results of the tests. The design validation was the description of expected behavior of the new implementation according to the prior testing results.

4. **Problem investigation may be problem-driven, solution-driven, goal-driven, or impact-driven**

The two main problem investigations were problem-driven (in the first cycle), due to the use of a problem scenario which had to be solved, and solution-driven in the second cycle by improving the implementation according to the test results.

5. **When designing a solution, maintain the design argument**

The stakeholder criteria are clearly defined in Chapter 1 and 2, these are seen as the validation of the design and are validated in the next section.

6. **When validating a design, consider trade-offs and sensitivity**

When trade-offs in the design were made, it was expressed clearly. Also the environment changes (given the test environment) were described clearly when occurred.

7. **When validating a design, aim to incorporate conditions of practice**

Throughout the research this was done in the sense that the testing goals were adapted to the prior knowledge of the testing participants. The experts had less experience in online chatting platforms, the students had less experience in Group Model Building in general.

8. **When solving a knowledge question in the regulative cycle by means of research, no research method is banned**

This was met in Chapter 2 when describing the foundations and context of Group Model Building, and linking this theory to innovative research according to a literature study.

With this, the design cycle is nearly completed. Some last elements are described in the next section. This will elaborate the final results of the test sessions and discuss the

implications it has. The last section (Section 8.3) describes the final conclusion of the test session and therefore concludes the second research question of this thesis.

8.2 Is the tool future proof?

Three tests have been conducted to see if the Group Model Building Bot in Slack forms a potential future for Distributed Group Model Building. During the testing processes, it became clear that the current "final" implementation is far from an actual final implementation. This is partly due to the technical problems that arose, but also due to some effects the Slack platform and the processes of the tool have on the quality of the Group Model Building session.

At first, it became clear that the participants missed face-to-face communication. Although being in the same room together, they tried communicating via the tool as much as possible. Their opinion clearly stated that this resulted in a chaotic manner of communication. Also text is in their opinion sometimes not enough to express an action you want to take or a comment you want to make.

This lack of face-to-face is however two-sided. Research by Hoppenbrouwers and Rouwette (2012) clearly shows that it is possible to conduct a sound Group Model Building session only by having a chat tool with clear openers. The lack of a face-to-face communication tool like Skype, was not seen as a pitfall. We have concluded that the openers are the cause of this. Another reason could be that the participants in Hoppenbrouwers and Rouwette (2012) were separated from each other during testing and had no other choice but to talk through the tool created. In other words prior to testing, they have setup their minds to communicate through the tool. Our Slack implementation used the same openers approach as Hoppenbrouwers and Rouwette (2012), but during testing participants were in the same room together, which might initiate the frustration because their natural sense wanted them to talk to each other face-to-face instead of through the tool (as this then is seen as an extra obstruction for communication). Nevertheless, the openers used were in the form of buttons that had to be pressed. After they were pressed the Bot asked questions to control the format of the answers given. This approach was clearly seen as frustrating and unnecessary complex. This resulted in the participants feeling not being heard, which neglects one of the goals of Group Model Building.

A second clear frustration was created by the Slack platform itself. This was caused by two things: First, having to switch between a bilateral conversation with the Bot and a normal conversation in the main channel, instead of having an overview of both at the same time was perceived as unworkable. Also not having a clear structured overview of all the possible threads created a chaotic overview. Therefore not being heard was not due to the implementation of the openers in the form of buttons, but due to the unnecessary number of actions that had to be taken in order to get some information in Slack.

Therefore in order to say if the Bot has potential, more research has to be done. What worked really well was the support of the Bot for the facilitator. But, for the facilitator the Bot acts more as an administrative aid instead of a communicative aid. Nevertheless, this created the possibility that the Bot can be used as an aid for a facilitator who is not that familiar with Group Model Building. This means in the future the Bot should be lesser in the role of a facilitator as it was in this research. On the other hand, the

Bot can be a full facilitator, but then it has to be a Bot which is equipped with a large artificial intelligence skills set.

So, is there thus a future for a Distributed Group Model Building Bot? The answer would be: Yes but not on a Slack platform and not as a partial facilitator. It would ideally be a structural and administrative aid for a facilitator not that familiar with Group Model Building on a platform like used in Hoppenbrouwers and Rouwette (2012). The only pitfall is that the chatting platform used by Hoppenbrouwers and Rouwette (2012) has been discontinued, therefore further research is difficult. Nevertheless it became clear that the future lays in such a tool with the preferable addition of a face-to-face communicative tool like Skype.

8.3 Evaluation conclusion

So in order to finalize this research, the second research question has to be answered. The evaluation question was:

Does a Distributed Group Model Building tool using Slack enable the benefits of regular Group Model Building?

The answer would be definitely not.

The structure of the Slack platform has its benefits but also clearly its pitfalls. The constant necessity to switch between different overviews made it impossible to create a complete overview of the information of the session. This resulted in frustration of the participants which potentially can lead to obstruction of creativity and therefore in the ability of solving a Messy Problem, thus, not committing in the goals and benefits of Group Model Building.

The bot itself has potential as was described in the previous section. The question if it should be a facilitator or just an aid of the facilitator remains. The latter seems workable but not on a Slack platform. What has been concluded is that the approach of Hoppenbrouwers and Rouwette (2012) can be extended with a Bot, but only if this is on a platform as open and structured as the one used in that research.

This concludes the final research question. The following chapter describes other possible routes that can be taken in order to improve the Distributed Group Model Building approach created in this thesis. It however must be made clear that further research is definitely necessary in order to create a sound distributed approach. Chapter 10 explains the overall conclusion of this thesis and describes the additional beneficial knowledge that has been created during this research.

Chapter 9

Further research

This chapter is about suggestions that can be made for further research. During the research of this thesis, a couple of considerations were made. These resulted due to complexity of implementation and time constraints. These considerations did result however in new ideas for implementation in the future. This chapter first talks about what is created already and then goes through some future research suggestions.

9.1 What has been produced?

In this thesis a description of two implementations of a Group Model Building aid that can be used in a distributed manner have been described. The second implementation was tagged as the final implementation of this research. It consisted of a Slack Bot that aids the facilitator and participants of a Group Model Building session. The benefits of using Slack was that it is a modern and popular chat tool that is already used in a lot of project settings.

The Slack Bot was built on a framework called BotKit which was written in NodeJS. The Bot is a listener and reacts when necessary. It therefore only acts when something is written down in the active chat session that fires a predefined statement inside the Bot's brain. This meant that the most difficult part of creating the Bot, was generating its brain. During the implementation of this thesis, approximately 4-5 months have been spent in developing this.

This meant some things were left out, or the developer seemed to be interested in developing but was outside the time constraints of this project. This resulted in a couple of considerations and pitfalls of the final implementation of this thesis. These were mentioned at the end of Section 6.2, but also arose during the testing of the implementations:

- Models of the causal loop diagrams still have to be drawn manually by the facilitator or a recorder.
- Adding and deleting variables is not possible during the causal relation creation phase.

- Sessions cannot be saved, in the sense that participants can continue on an old session.¹
- The bot is still not very smart and does not have any people skills.

These considerations (or pitfalls) resulted in the research suggestions that are described in the following sections.

9.2 Integrating model generation

The first pitfall we described above was: *Models of the causal loop diagrams still have to be drawn manually by the facilitator or a recorder.* In the current implementation this can be facilitated if the leading user on his computer draws the causal loop diagram and posts pictures in the chat of the main channel. But this is highly unfeasible given the time it would take and the potential speed of the chat and the creation. Therefore this is highly unwanted.

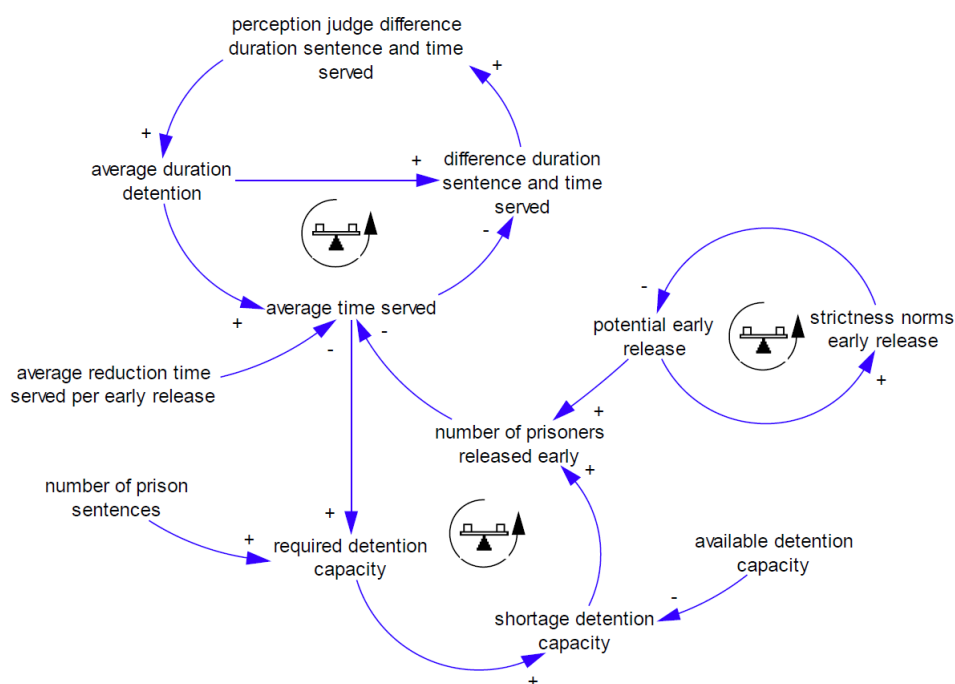


FIGURE 9.1: Example causal loop diagram (system dynamics) on early release policy (Rouwette et al., 2007).

The addition of automated model generation would be the answer. Given the potential of Slack and the BotKit framework, this can be done but takes time. All the linked variables are formally structured in the program code. This means that it is easy to write an extra API that can communicate with a drawing tool that automatically posts images of the model created thus far.

¹The produced results however remain in the channel as chat messages, and are therefore this approachable.

If we look again at the example causal loop diagram used in Rouwette et al. (2007) (Figure 9.1), we can see that it is simply an automated placements of nodes connected with edges (van Bommel, 2017). Writing such an automated tool would not be that difficult. Its product can than be saved on the dedicated NodeJS web server which than automatically is loaded by the Group Model Building bot.

That would be a possible solution to create causal loop diagram automatically which follows live the creation of the relations in the Group Model Building session in Slack. However this seems feasible, it should be researched if such an implementation is useful within the chat structure that we have to cope with and will not create an opposite goal.

9.3 Central flexibility

The second pitfall was: *Adding and deleting variables is not possible during the causal relation creation phase.* The final implementation described in this thesis had not implemented this due to technical limits. The whole control panel that was sent to the leading user in order to coordinate the create causal relations phase had to be a predefined set of variables and constraints. Changing something, like for instance adding a variable, would not influence the control panel, or a completely new control panel had to be sent.

Given that it is not feasible to have more than one control panel in the bilateral conversation with the Group Model Building Bot, it was chosen to separate the adding and deleting from the causal creation phase. Of course in a natural Group Model Building session this would obstruct creativity because not enough insight in the model can be generated.

Future research should find out if it is possible to create a handy way of adding, changing and deleting variables that are digitally already been trough calculation. This would mean that the overview message, control panel and potential created diagram have to be updated as well.

The third pitfall was: *Sessions cannot be saved, in the sense that participants can continue on an old session.* This is highly necessary in normal condition. As described in a lot of prior research (Rouwette et al., 2002) (Rouwette et al., 2007) (Rouwette and Franco, 2014), Group Model Building sessions are never on their own but are part of a larger set of sessions in which the models and diagrams are constantly looked at again and again to improve and change them. Therefore, saving sessions in Slack so that participant can reload their old sessions including intermediate results.

The suggestion here would be that research should be done in finding a suitable way to do this. Were are the sessions saved? What is saved and what not? Who can reopen saved sessions? What can be adjusted in saved sessions? These are all questions that have to be looked into.

It all comes down to central flexibility. If the above is possible, switching between the different Group Model Building phases should be possible as well. The final implementation of this thesis has left it as "out of scope" due to complexity and time constraints, but future research should improve and extend the Distributed Group Model Building capabilities.

9.4 An AI facilitator

The last pitfall and probably the most interesting was: *The bot is still not very smart and does not have any people skills*. The goal of this thesis was to create a Distributed Group Model Building application that would make it more easy to use for a larger audience. Although prior Group Model Building skills are not required, with the exception of finding feedback loops, there is still relied highly on the facilitation skills (Schein, 1990) of the facilitator at hand.

The next step would be a full artificial intelligence Bot that not only acts as an aid for the process, but also acts as a full facilitator. This would mean that the brain of the Group Model Building Bot would be extended by machine learning and artificial human interaction skills.

A lot of prior research has been done in artificial intelligence and team collaboration, like Orwig, Chen, Vogel, and Nunamaker (1997), Soller (2001), Aronson, Liang, and Turban (2005), and Wenger (2014). An artificial bot that acts as a natural human being would completely remove any required prior knowledge for using the Distributed Group Model Building tool. In the end, even a facilitator in the form of a user would be unnecessary.

These four suggestions define an improved future for a Distributed Group Model Building tool. Although we acknowledge that this is a long way from now, it can in the end be the answer to a collaborative tool that can be used in a distributed setting, without any prior knowledge inside an already popular chat system.

Chapter 10

Conclusion

In this thesis a tool has been created that potentially could be used as a Distributed Group Model Building approach. It was based on previous research by Ouborg (2015) and extended with theory as described in Chapter 2. In general, Hoppenbrouwers and Rouwette (2012) was seen as the main inspiration to extend the research by Ouborg (2015). In order to achieve this, a combined approach was created and implemented as a chat bot on the Slack platform. The bot functions as an aid for the facilitator and in some cases as the facilitator itself.

Two implementations have been made, the first was simply a translation of Ouborg (2015) into a Slack Bot. This implementation was tested with the help of a group of students. The second was an improved version of the first, given the feedback that was acquired in the test sessions. This resulted in the second and final implementation of this thesis in which more or less the combined approach was implemented. This combined approach was in return tested with the same group of students and with a group of Group Model Building experts.

These tests showed that the final implementation did not function as a sound approach to Distributed Group Model Building. This was partly due to the implementation itself and partly due to the platform it was created on. This did not mean however that there is no potential future for a chat bot in helping creating a Distributed Group Model Building approach. The conclusion is that a platform like Hoppenbrouwers and Rouwette (2012) should be used, with the extension of a bot acting as a structuring aid for the facilitator. This creates the possibility of having a distributed Group Model Building approach that does not require prior knowledge of the Group Model Building technique itself.

Last, extra suggestions were given in order to improve the chat bot approach. In the end it is clear that a bot has potential, but not on a Slack platform with an implementation created in this thesis. This thesis however has showed that when some previous research approaches are combined, a better basis for a Distributed Group Model Building approach is created. It is now up to future research to see what the balance should be between a Group Model Building bot and the facilitator. The future for a distributed approach in Group Model Building remains hopeful, which means even more organizations in the end will be able to solve their Messy Problems in a distributed and more agile manner.

Bibliography

- Philip Achimugu, Ali Selamat, Roliana Ibrahim, and Mohd Naz'ri Mahrin. A systematic literature review of software requirements prioritization research. *Information and Software Technology*, 56(6):568 – 585, 2014. ISSN 0950-5849. doi: <https://doi.org/10.1016/j.infsof.2014.02.001>. URL <http://www.sciencedirect.com/science/article/pii/S0950584914000354>.
- R. L. Ackoff. *Redesigning the Future: a Systems Approach to Societal Problems*. Wiley, New York, 1st edition, 1974.
- Russell L Ackoff. The future of operational research is past. *Journal of the operational research society*, pages 93–104, 1979.
- Katie Elson Anderson. Getting acquainted with social networks and apps: picking up the slack in communication and collaboration. *Library Hi Tech News*, 33(9):6–9, 2016. doi: 10.1108/LHTN-10-2016-0049. URL <http://www.emeraldinsight.com/doi/abs/10.1108/LHTN-10-2016-0049>.
- Jay E Aronson, Ting-Peng Liang, and E Turban. *Decision support systems and intelligent systems*. Pearson Prentice-Hall, 2005.
- Saniye Tugba Bulu and Susan Pedersen. Scaffolding middle school students' content knowledge and ill-structured problem solving in a problem-based hypermedia learning environment. *Educational Technology Research and Development*, 58(5):507–529, 2010. ISSN 10421629, 15566501. URL <http://www.jstor.org/stable/40929416>.
- G-J De Vreede and Robert O Briggs. Collaboration engineering: designing repeatable processes for high-value collaborative tasks. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 17c–17c. IEEE, 2005.
- A. Delbecq, A. Van de Ven, and D. Gustafson. *Group techniques for program planning a guide to nominal group and delphi processes*. Scott Foresman Company Glenview, Illinois, 1975.
- Fast Company. Why slack is one of the most innovative companies of 2017, February 2017. URL <https://www.fastcompany.com/3067475/why-slack-is-one-of-the-most-innovative-companies-of-2017>. [Online; posted february-13-2017, written by Fast Company Staff].
- J. W. Forrester. *Industrial dynamics*. Productivity Press, Cambridge, MA, 1961.
- J. W. Forrester. *Industrial dynamics - after the first decade*. Pegasus Communications, Williston, VT, 1975.

- Alan R Hevner. A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4, 2007.
- Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004. ISSN 02767783. URL <http://www.jstor.org/stable/25148625>.
- Stijn Hoppenbrouwers and Etiënne Rouwette. A dialogue game for analysing group model building: framing collaborative modelling and its facilitation. *International Journal of Organisational Design and Engineering*, 2(1):19–40, 2012.
- Stijn Hoppenbrouwers and Wim van Stokkum. From dialogue games to m-thinklets: Overview and synthesis of a collaborative modeling approach. *International Journal of e-Collaboration (IJeC)*, 9(4):32–44, 2013.
- Stijn Hoppenbrouwers and Ilona Wilmont. Focused conceptualisation: framing questioning and answering in model-oriented dialogue games. *The Practice of Enterprise Modeling*, pages 190–204, 2010.
- Peter S Hovmand, David F Andersen, Etiënne Rouwette, George P Richardson, Krista Rux, and Annaliese Calhoun. Group model-building ‘scripts’ as a collaborative planning tool. *Systems Research and Behavioral Science*, 29(2):179–193, 2012.
- Juhani Iivari. A paradigmatic analysis of information systems as a design science. *Scandinavian journal of information systems*, 19(2):5, 2007.
- Nurul Jumaat and Zaidatun Tasir. Instructional scaffolding in online learning environment: A meta-analysis. pages 74–77, 04 2014.
- Carlene Lebeuf, Margaret-Anne Storey, and Alexey Zagalsky. How software developers mitigate collaboration friction with chatbots. *arXiv preprint arXiv:1702.07011*, 2017.
- Allen Marshall, Rose F. Gamble, and Matthew L. Hale. Outcomes of emotional content from agile team forum posts. In *Proceedings of the 1st International Workshop on Emotion Awareness in Software Engineering, SEmotion '16*, pages 6–11, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4169-1. doi: 10.1145/2897000.2897003. URL <http://doi.acm.org.ru.idm.oclc.org/10.1145/2897000.2897003>.
- Stefan Oppl. Articulation of work process models for organizational alignment and informed information system design. *Information & Management*, 53(5):591–608, 2016.
- R Orwig, Hsinchun Chen, D Vogel, and Jay F Nunamaker. A multi-agent view of strategic planning using group support systems and artificial intelligence. *Group Decision and Negotiation*, 6(1):37–59, 1997.
- A. F. Osborn. *Applied imagination*. Charles Scribner’s, Oxford, UK, 1st edition, 1953.
- A. F. Osborn. *Applied imagination*. Charles Scribner’s, New York, 2nd edition, 1957.
- A. F. Osborn. *Applied imagination*. Charles Scribner’s, New York, 3rd edition, 1963.
- M.J. Ouborg. Formalizing group model building; the next step in supporting group decision making. Bachelor thesis, Radboud University of Nijmegen, 2015.

- G. F. M. Paulussen and Th. P. Van der Weide. *Domain Modeling: The systematic construction of an ontology*. 2007.
- Jeffrey M. Perkel. How scientists use slack. *Nature*, 536:285–291, 2016.
- Andrew Perrin. Social media usage. *Pew Research Center*, 2015.
- Andrew Ravenscroft and Simon McAlister. Designing interaction as a dialogue game. *Interactions in Online Education: implications for theory and practice*, page 75, 2006.
- George P Richardson and David F Andersen. Teamwork in group model building. *System Dynamics Review*, 11(2):113–137, 1995.
- Horst W. J. Rittel and Melvin M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4:155–169, 1973.
- Stephen P. Robbins and Neil Barnwell. *Organisation Theory: concepts and cases*. Pearson Education, Australia, 5th edition, 2006.
- E. A. J. A. Rouwette and L. A. Franco. *Messy problems: Practical interventions for working through complexity, uncertainty and conflict*. Radboud University, Nijmegen, 2nd edition, 2014. Course material from Intervention Methodology MAN-BCU019.
- EAJA Rouwette, JAM Vennix, P van Hooff, and Wouter Jongebreur. Modeling crime control in the netherlands: insights on process. 2007.
- Etiënne AJA Rouwette, Jac AM Vennix, and Theo van Mullekom. Group model building effectiveness: a review of assessment studies. *System Dynamics Review*, 18(1):5–45, 2002.
- Edgar H Schein. *Process consultation. Vol. 2, lessons for managers and consultants*. Addison-Wesley, 1987.
- Edgar H Schein. A general philosophy of helping: Process consultation. *MIT Sloan Management Review*, 31(3):57, 1990.
- Edgar H Schein. The concept of “client” from a process consultation perspective: A guide for change agents. *Journal of Organizational Change Management*, 10(3):202–216, 1997.
- Amy Soller. Supporting Social Interaction in an Intelligent Collaborative Learning System. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12:40–62, 2001. URL <https://telearn.archives-ouvertes.fr/hal-00197321>. Part II of the Special Issue on Analysing Educational Dialogue Interaction (editor: Rachel Pilkington).
- Wolfgang Stroebe, Bernard A Nijstad, and Eric F Rietzschel. Chapter four-beyond productivity loss in brainstorming groups: The evolution of a question. *Advances in experimental social psychology*, 43:157–203, 2010.
- The Atlantic. Here comes the slack backlash, April 2016. URL <https://www.theatlantic.com/technology/archive/2016/04/i-love-slack-i-hate-slack-i-love-slack/479145/>. [Online; posted april-20-2016, written by Adrienne Lafrance].

- Patrick van Bommel. *Foundations of information systems*. Institute of Computing and Information Sciences, Radboud University, Nijmegen, 2017.
- Bas van Gils and Dovidé Vojevodina. The effects of exceptions on enterprise architecture. *Technical Report; ICIS-R06011*, 2006.
- Jac A. M. Vennix. *Group Model Building; Facilitating Team Learning Using System Dynamics*. John Wiley & Sons Ltd., Sussex, England, 1st edition, 1996.
- Etienne Wenger. *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann, 2014.
- Mathias Weske. *Business process management: concepts, languages, architectures*. Springer Publishing Company, Incorporated, 2010.
- Roel Wieringa. Design science as nested problem solving. In *Proceedings of the 4th international conference on design science research in information systems and technology*, page 8. ACM, 2009.