Developing a Formal Framework for Simulation Games

Cecile H.F.M. Crobach
Faculty of Technology, Policy and Management, Delft University of Technology
Jaffalaan 5, 2628 BX Delft, The Netherlands

Abstract
Simulation gaming (SG) is a method that is increasingly used to improve business processes in various organizations. Currently no formal framework for SG exists, though having such a framework can have a lot of advantages. This paper describes the development of a formal framework for rigid-rule SG, which is done in a deductive and inductive way; it is based on theory and on knowledge gained during a case study that was executed at a large Dutch bank. The result is a framework consisting of three main components; a generator, a transducer, and an acceptor, each with a list of property requirements that they should meet. The framework should be put to practice in order to be further developed and to find out whether statements made about its advantages and applicability are true.

Keywords: Simulation Gaming, Formal Framework, Improved Development of Simulation Games, Rigid-rule Games.

1. INTRODUCTION
As the focus of organizations on improving business processes has increased significantly in the last few years, the interest in and use of different analysis methods has grown. Companies use pilots, simulations, games, and many more methods in order to improve their practices. The popularity of simulation games (SG) is increasing and the method is used in many different cases and for numerous purposes. Currently, no formal framework for SG exists, though such a framework can have many advantages; it would enable an easier assessment of the educational effectiveness, provide a proper base for collaboration between game developers, allow game designers to add educational content more easily and efficiently, and increase the educational effect of simulation games. Harteveld (2009) also identified the need for a theoretical underpinning of games. He states that "*games are studied without a proper framework onto which the results can be reflected on*" (Harteveld, 2009, p. 1).

During a literature search no clear or suitable guidelines for developing a formal framework for SG were found, so another starting point is selected; the framework for modeling and simulation (M&S) (Zeigler, Prachofer, & Kim, 1976). This framework is chosen as a starting point for developing the SG framework, because M&S is closely related to SG; in both methods, for example, first a simplified model of reality is made, this model is used to learn about the current and future situation, and finally the gained knowledge will be translated back to reality (Peters, Vissers, & Heijne, 1998; Verbraeck & Valentin, 2006). In order to properly adjust this framework, both a deductive and an inductive approach are taken. The deductive approach includes analyzing the M&S framework and comparing it to the components of a SG. The inductive approach entails executing a case study and using findings from the case study to base the framework on.

Games can either be open (free-form games) or closed (rigid-rule games) (Klabbers, 2003). In rigid-rule games the designer has all the necessary information as the process or the system is well-defined. In open games, however, the system is so complex that it is not possible to define everything in the game. The case study that is executed is based on a closed model, and therefore will result in a rigid-rule game. Based on the presented information, the following scientific research question and sub-questions will be answered in this article:

How can the modeling and simulation framework be adjusted in order to develop a formal framework for rigid-rule simulation games?

a. What does the formal framework for modeling and simulation look like?
b. What are components and characteristics of SG?

In order to answer these questions, the paper is structured as follows. Section 2 will describe the theoretical foundation needed to develop the framework. Based on the theoretical foundation and the knowledge gained from the case study, section 3 describes the development of the framework. Section 4 will verify and evaluate the framework, and section 5 will present the conclusions and a discussion on the findings of the article.

2. THEORETICAL FOUNDATION

2.1 M&S Framework

Zeigler et al (1976) developed a formal framework for M&S as shown in Figure 1. It shows the basic entities of M&S and their relationships.

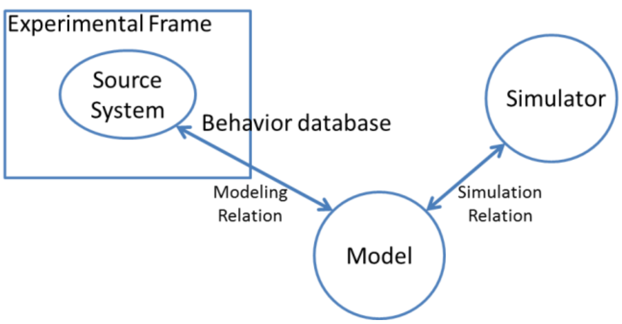

Figure 1 Basic entities and their relationships in M&S (Zeigler)

The source system is the real (or virtual) environment that is being modeled. Data that is gathered by observing or experimenting with the source system is termed the behavior database. A model, according to Zeigler et al (1976) can be seen as "*a set of instructions, rules, equations, or constraints for generating input and output behavior*" (p29). It is "*a state transition and output generation mechanism in order to accept input trajectories and generate output trajectories depending on its initial state setting*" (p30). The modeling relation shows the transformation from the source system and its experimental frame to the model, and the simulation relation shows the conversion from the model to the simulator. The simulator is a device for executing the model.

The source system, however, cannot be modeled apart from its context, which is why a model is given an experimental frame (EF) to represent the context of the system in the real world. This EF specifies the conditions under which the system is observed and experimented with (Zeigler, Prachofer, & Kim, 1976). Based on Wymore's general systems theory (Wymore, 1977) Zeigler developed an experimental frame (EF) that can be defined as: "*The specification of circumstances under which a model (or a real system) is observed and experimented with*" (Zeigler, 1984). The EF serves three important purposes. First, the frame generates model inputs. Second, the reaction of the model on inputs is observed and the data is collected. Third, it controls the experimentation as it puts constraints on values of the state variables of the model.

The EF consists of three subparts (Figure 2): a generator, an acceptor, and a transducer. The generator is used to create the set of admissible input segments; it produces input segments for the model. Input segments consist of information that varies in time, like periodic arrivals, stochastic arrivals, or workload characteristics.

The EF selects the system outputs of interest through an acceptor. The system is naturally fed with real data, and the EF has to select those inputs that are of interest (Traoré & Muzy, 2006). The acceptor also monitors an experiment to see if the desired conditions are met; it continually tests the run control segments to see if they meet the given constraints; values that meet the experimentation constraints are accepted, others are not. Examples of run control segments are the terminating conditions of a system run, the startup period, and the number of replications.

The transducer observes and analyzes the system output segments; it collects the input/output data and summarizes the records. Typical transducer information is turnaround time or failure rate. Output segments can for example be entities or information on a resource.

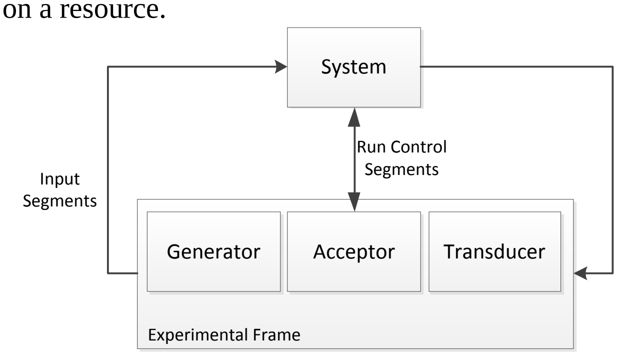

Figure 2 Experimental Frame Structure (Zeigler, Prachofer, & Kim, 1976, p. 27)

Next, information is needed on the components of a SG that need to be added to the M&S framework.

2.2 Simulation Games

The working definition of a SG according to Klabbers (2006) is:

"A game is any contest or effort (play) among adversaries or teammates (players) operating under constraints (rules and resources) for an objective (winning, victory, prestige, status, or pay-off). The exercise or activity should involve overt competition or cooperation between the individuals or teams, who are competing against each other or together (while jointly conquering circumstances) fighting the odds" (Klabbers, 2006, p. 28).

Related to this definition Klabbers (2006) states that a game has three main components: rules, resources, and players. Figure 3 depicts the basic architecture of games and simulations with its three components and their relations. The actors establish systems of interactions and they use rules and resources while they function in that system. Klabbers (2006) gives the example of a soccer game, in which the players, the coaches, and the referees are the main actors and these actors interact based on the rules. The ball and the field are examples of resources they use. The reference system, which is the sets of resources, can be seen as the game space; it is the set of places for resource allocation. This reference system symbolizes the physical environment and the infrastructure of a real or imaginary world (Klabbers, 2006).

Based on the working definition given above, the applications, components, and typology of SG are further researched. Information of several scholars is used to gather knowledge on those issues. Several authors mention the importance of debriefing; knowledge results from the combination of grasping and transforming experience, debriefing is necessary as it is the transformation of the game experience into learning (Kolb & Kolb, 2005; Kolb D. , 1984). Also, debriefing is needed in order to generalize learnings together with the other players (Crookall, 2010). Different sides of the three components are mentioned, for example that players used to be only real people, but virtual characters are being used more often in games as well (Faria, Hutchinson, Wellington, & Gold, 2009). The different types of SG are role-plays, card games, computer simulations, and interactive games (Caluwé, Hofstede, & Peters, 2008). Finally the approach of developing a SG is discussed by several scholars (Duke, 1974; Duke & Geurts, 2004; Wenzler, 1997).The main conclusions that are drawn on the findings of all of these scholars are: the three interconnected parts of players, rules, and resources together with the activities taking place in a game should in some way be incorporated in the formal framework. The players ought to, based on the results of their activities, rethink and adjust their actions. Besides that, it is important that the framework is able to capture different types of games; computer-based or non-computer-based, and games with real players, virtual players, or both.

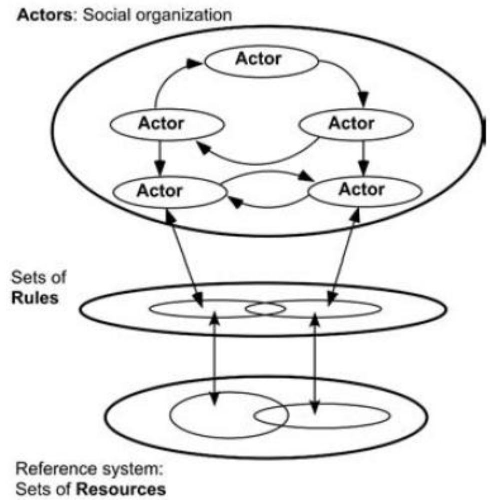

Figure 3 Generic model of games and simulations (Klabbers, 2006, p. 42)

Next, an inductive approach to develop the framework is taken. A description of the case study is given in the information block. The following section describes the SG that is developed in the case study and the comparison of the M&S framework and the developed SG.

3. DEVELOPING THE FRAMEWORK THROUGH AN INDUCTIVE APPROACH

When comparing the formal framework of M&S to the components of a SG that should be added to it, some issues arise. First of all, in the M&S framework, no automatic feedback loop is apparent between the components of the EF. In a SG, however, the feedback loop is one of the most important characteristics as there is a continuous cycle between the system and the player. Therefore, some way should be found to link the components of the EF in the framework for SG, so a cycle can be produced. The players, for example, should be able to influence the EF, but they should also be able to gather information from the EF. As for the rules, there should be a component that stops the player if he is not complying with the rules, or that limits the behavior of the player so that he has no choice but to comply with the rules. However, in the M&S framework, only the acceptor determines which data

Case Study

A large Dutch bank wants to decrease the handling time of complaints. Management has selected several alternatives to achieve this goal, but before choosing whether these options should be implemented or not, more information is needed on the consequences of the alternatives. Implementing them may not only have an influence on the throughput time of complaints, but it will also have an influence on the working activities of the employees. It is important to increase the insight and the acceptance of employees for the different alternatives because else they may prevent a successful implementation. The practical research question and its sub questions were posed as follows:

How can a Simulation Game be used to improve business processes?
a. Can a SG be used in order to decrease the throughput time and pickup time of handling complaints?
b. Can a SG be used in order to increase the insight of employees for the different alternatives?

DES enables modeling and analyzing the behavior of complex systems, and it allows making quick adjustments in the modeled system so alternatives can be tested and analyzed in a short period (Verbraeck & Valentin, 2006). As DES is able to deal with entities (single requests), different resources used by a process, and the stochastic nature of the process (Nidumolu, Menon, & Zeigler, 1998; Kampen, 2006), the method is very suitable to model and simulate the business process of handling complaints. However, as the project also has qualitative goals, using only DES would not suffice. Therefore, a SG based on DES was developed.

The main conclusion of the case study was that the SG based on DES can be used to analyze what the effect of different alternatives is on the throughput time, pickup time, and other indicators. The bank is recommended to implement one of the tested alternatives, which will decrease both the throughput and pickup time.

The second sub question was answered by using interviews and questionnaires before and after the SG sessions with the employees. In general, the participants stated that they gained insight in the results of each alternative and based on that their opinion on implementing the alternatives got more positive. Besides these insights, the employees gave a lot of useful comments for the implementation of the alternatives.

is accepted into the model. It does not have to pause the whole system, for example, because the inputs are given at once to the system, and not in phases as can happen during a SG. When considering the resources, in the M&S framework, during the run no extra event cards for example enter the system. The creation of other resources should be added to the framework; this may be done by the generator, but it may also be done by the transducer.

As can be seen, several issues arise when the M&S framework is adjusted in order to contain the SG components. The inductive approach will help get a better grip on these issues..

3.1 Description Service Game

One of the major components of the Service Game, which is the name of the developed SG, is the player. This is a real player, and as described before, the only information known about the player are the inputs and outputs; the player is a black box. Based on several information sources the player will crack his brain and decide what decisions he will make. The player can enter his decisions in an interface that is developed in a spreadsheet. The player can choose and differentiate the values of six variables. The chosen values for those variables are entered into the DES model. The DES model runs, and will show the information of two indicators during the run in a spreadsheet. Based on this information, the player will base his next decisions and the cycle will repeat itself. Each cycle takes 2 weeks in virtual time; after those 2 weeks the simulation model pauses, presents the player with the information, and eventually the player will decide how to act upon this information and continue the model run again for 2 weeks.

The information that is produced during each run in the spreadsheet can be seen as input segments for the player; based on this information, the player decides in the black box what his outputs will be. These “outputs” of the player, are the decisions he makes.

3.2 Linking Description and M&S Framework

When mapping all this information to the terms used in the M&S framework, the player can be seen as the system, and everything else can be seen as part of the EF. This mapping follows logically from the tasks of the components and the role of the player in the SG; the player is the only part of the SG with an unknown internal structure, and the other parts of the SG have tasks that all met the task description of the EF components.

The EF has three major tasks. When these general tasks are specified for the Service Game they can be stated as follows:

1. Generate input segments for the player (the spreadsheet and the schedule information in the Service Game)
2. Control the experimentation by putting constraints on the output segments; by using the spreadsheet, no wrong inputs can be entered into the model.
3. Observe and analyze the system output segments (the player), collect the data and summarize the records (the DES model uses the accepted data and computes the model based on that).

When further deciphering the tasks undertaken in the SG, the first task of generating the inputs for the player is partly done by the spreadsheet, and partly by the facilitator; even though the DES model contains the actual simulation processes, the spreadsheet translates the information to a format so the player can interpret the information. The facilitator produces extra information, like event cards or in the case of the Service Game the working schedules. Together, they provide the player with the necessary information during the run. As in the M&S framework the task of the generator is generating the input segments for the system, the major generator in the SG is the interface between the computational model and the player.

The second task is in this case done by the spreadsheet, as no wrong data inputs can be entered in the spreadsheet. Another controlling task that is done, however, concerns the timing of the game rounds. The timing of these rounds is, in this case, controlled by the DES model itself; it pauses after each two weeks, and continues whenever the player has entered his new decisions. The information concerning the run set up of the game can be compared to the run control segments of the M&S framework; the run control segments are the run set-up variables as entered into the simulation model. The acceptor will continuously check if the two weeks, for example, have already passed, and will then pause the model as stated by the run control segments. In other games, however, the run length can also be terminated by the facilitator or by a certain event happening in the game. These are all different options, but they should be controlled and monitored. The task description given by Zeigler et al (1976) that matches with this information is that of the acceptor; the acceptor determines which data to accept and monitors if all conditions are met. In terms of the SG, the acceptor should also monitor if all the rules of the game are complied with; are the correct actions taken by the players, does the model pause at the right times etc. Whether the right actions are taken can also be ensured by designing the interface between the player and the experimental frame in such a way that no unallowable values can be entered into the model.

The third task of observing and analyzing the system outputs and collecting the data is in this case done by the DES model. It only observes and analyzes the outputs that have been accepted by the acceptor, and send through to the DES model. The choices made by the player as entered into the spreadsheet are translated into variables and processes in the simulation model. The task description of a transducer is analyzing the system output segments and summarizing them. This description matches with the tasks of the DES model. This comparison is shown in Figure 4.

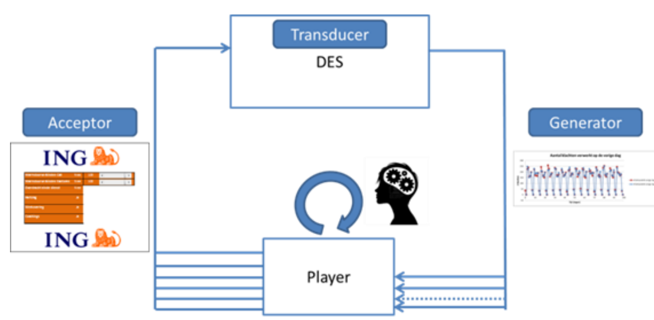

Figure 4 Schematic Serivce Game

4.3 Requirements of Formal Framework

A list of property requirements for the acceptor, transducer, and generator in a SG is developed:

Acceptor:

- Must be able to gather the output segments of the player (the player's behavior)

- Must be able to control the output segments of the player
  - Must be able to report unpermitted output segments back to the player, and/or
  - Must be able to make sure that the player cannot produce unpermitted output segments
- Must be able to send accepted output segments of the player to the transducer
- Must be able to control the run control segments as demanded by the game
  - Must be able to run the model until the facilitator stops it, and/or
  - Must be able to run the model for a certain, predetermined period, and/or
  - Must be able to run the model for a certain period until an event takes place, and/or
  - Must be able to restart the model when the facilitator starts it, and/or
  - Must be able to restart the model when a certain event has occured
- Must be able to make sure the rules of the game are followed

Transducer:
- Must be able to gather the output segments that are accepted by the acceptor.
- Must be able to execute the entire model (function F) based on a certain starting situation and influenced by the outputs of the acceptor.
- Must be able to remember historic values, as the start state after each round is the beginning state of the next
- Must be able to analyze the output segments of the acceptor and the results of them on the model
- Must be able to summarize the results.

Generator:
- Must be able to gather data as produced by the transducer.
- Must be able to translate the data from the transducer to the determined output variables.
- Must be able to present input segments to the player
  - In a continuous fashion and/or
  - After a specified period (i.e. after a few rounds the player gets more information)
- Must be able to produce information itself, i.e. that does not come from the transducer

This list of properties can be met in different ways. The facilitator, for example, might take up some tasks of the different components. The facilitator may take up a task of the acceptor component if he makes sure the rules of the games are complied with. If the facilitator is given the task of producing the event cards, for example, he will take up part of the generator requirements. It is therefore not necessary that there is one fixed component that meets all the acceptor requirements, for example, but it may be that different components of a SG, including the facilitator, together meet all the requirements.

4. VERIFICATION AND EVALUATION OF THE FRAMEWORK

4.1 Verification

It may seem as if the model and the simulator part of the M&S framework have been left out in the SG framework. This is caused by the fact that in the case study a DES model is used in which the simulator is already incorporated in the software that is used to build the model. Therefore, it is not 100% comparable to Zeiglers framework in which there is a clear separation between the model and the simulator. The simulation relation as shown in the M&S framework is therefore not apparent in that same way in the SG framework. The M&S framework in theory differs from the application in practice; the model and simulator are both incorporated in the same software package. The source system is the CKA department, which is modeled with its experimental frame in the DES model of the Service Game. The model is developed in the program Arena, which already includes the simulator. As explained above, the DES model developed in Arena is actually part of the transducer; the transducer must be able to execute the entire model (function F) based on a certain starting situation and influenced by the outputs of the acceptor. Therefore, it can be stated that the simulator and model and their simulation relation are apparent in the SG framework, although depicted in a different way than in the M&S framework.

As shown in Figure 5, the M&S framework actually also exists within the player block; the source system is a Unit Manager of the CKA department, and this role is modeled and simulated by the player of the SG.

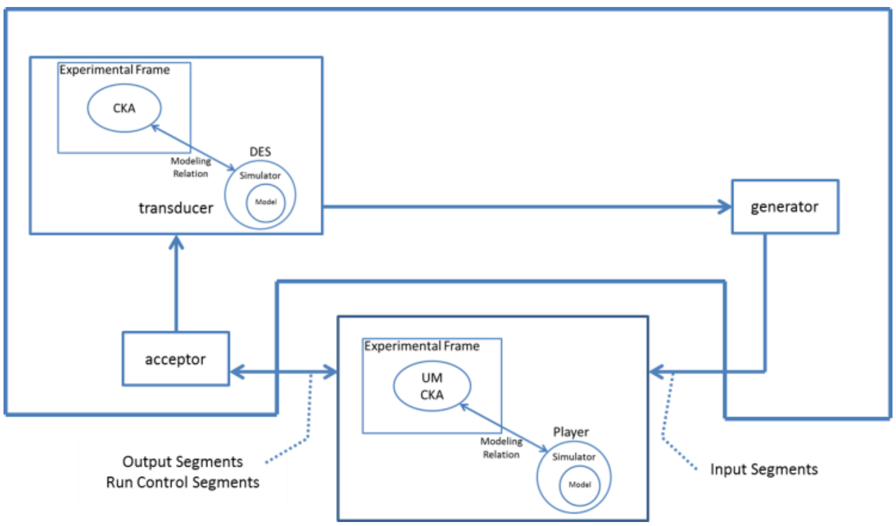

Figure 5 Model and simulator in the SG framework

4.2 Evaluation

Three experts are interviewed to evaluate the framework. First, Harald Warmelink, assistant professor at Delft University of Technology stated that the framework has added value and that it can be used for the following purposes. It can be used:

- at the beginning of the design phase in order to see whether a certain problem can be approached with the selected game engine.
- when a first concept of a game has been designed, in order to check what parts of the SG are still missing.
- as a framework to implement a concept as it clearly shows what relationships need to be present in the SG, and what the components of the SG need to be able to do.
- during the design phase to find out if the SG will be able to do what the designer wants it to do; is it able to transfer a certain function F while using inputs of and producing outputs for the player?
- to think about how the simulation game concept can be implemented.

The framework also helps those people who do not have a lot of experience with computational models like DES as this framework can be used to think about how the simulation game concept can be implemented. Furthermore, using this framework will enable improved communication between designers of a game or between the designer and the client; they will know exactly what part of the game they are talking about, making the design more concise and it would decrease ambiguity of games that currently often exists (e.g. already in the definition of simulation games).

The framework also enables an easier assessment of the educational effectiveness of SG as it systematically describes the objective and the components that should reach the objectives of the SG (Warmelink, 2012).

Another expert interview, held with Sebastiaan Meijer, assistant professor at Delft University of Technology, showed that what Warmelink said is true, but that in order to really reap these advantages and the added value of the framework, it should be further developed. It was also stated that the usability of the framework should be seen mostly for gaming simulations, i.e. the more computational games (Meijer, 2012).

The final interview is held with Jan Klabbers, whose work is an important foundation for this article. The most important thing that he pointed out is that the framework is very valuable, but indeed only for closed games. The framework will probably not work for open games. The stated advantages of a formal framework are indeed true for this framework, and that the framework is also applicable to other types of games as well (board games or role-plays) as long as they are rigid-rule games (Klabbers J. , 2012).

He also mentioned that in this framework, the player is not classified, although different types of classifications of players exist, and these classifications have an impact on their behavior. If this classification would be taken into account, the system would switch from a referential system to a self-referential system. Self-referential systems also take the social and human dimension into account whereas the referential system mostly focuses on the natural environment and the technology domain (Klabbers, 2006, p. 153). In other words, in the framework the player is seen as a black box, and the characteristics of the potential players in the transducer are not discussed either. The developed framework for SG works for referential systems, and may be adjusted to incorporate self-referential games as well, but how this can be done should be further researched.

There are some disagreements between the experts on the application of the framework. This is material for future research.

5. CONCLUSIONS AND DISCUSSION

The result of this paper is a framework consisting of three main components; a generator, a transducer, and an acceptor. For each of these components, a list of property requirements is made which lists all the demands the components should meet.

In the introduction several advantages of formalization were posed. It was stated that formalization would enable improved communication between designers of a game. The expert evaluation indeed shows this advantage of the formal framework; it can be used during discussions with designers, for example, to make their discussion more structured, causing the ambiguity that currently exists in the area of SG to decrease. Next, it was stated that the framework would enable expressing the simulation game apart from the implementation; it would enable an analogical view of the model and make the game more generalized, making replication of the game easier. The framework did indeed enable expressing the SG apart from the implementation; the components and relationships can be identified and analyzed apart from the actual implementation. Whether it is easier to replicate a SG when using the formal framework should be researched further; there is not enough evidence to acknowledge this statement. Furthermore, it was stated that it would enable an easier assessment of the educational effectiveness, would allow game designers to add educational content more easily and efficiently, and would increase the educational effect of simulation games. Klabbers (2012) confirmed these advantages, as long as a rigid-rule game is concerned.

During the expert evaluations some disagreements arose on the applicability of the framework. Therefore, the framework is applied to a board game and to a role-play (Crobach, 2012). Based on this application, it seems that the framework is also applicable to board games and role-plays, as long as they are rigid-rule games of closed models. This is also what Klabbers stated during the expert interview. However, in order to further confirm this statement the framework should be applied in order to design a board game or role-play.

In order to provide more evidence on the purposes, the applicability, and the advantages of this framework, it will be necessary to apply the framework in different cases, so the claimed advantages and applications can be tested.

Several other areas of future research are identified. The first area for future research was identified by Klabbers (2012). As this framework was developed for rigid-rule games, it probably will not be applicable to open games. It should be researched how this framework can be adjusted so that it can be applied to open games as well. The developed framework for SG works for referential systems, and may be adjusted to incorporate self-referential games as well, but how this can be done should also be further researched (Klabbers, 2012). This relates to the following area of future research:

During this thesis, the SG did not have the goal of analyzing and getting insight in the system of the player itself, only influencing it. However, it would be very intersting to try and map the entire player. This can be linked to Klir's levels of system knowledge (Klir G. , 1985). Klir stated there are four levels of system knowledge. The first level is the source system, at which it is known what variables should be measured and how they should be observed. The second level is the data level, at which data is collected from a source system. The generative level is the third level, at which data is generated. The fourth and final level is the structure level, at which all components are coupled together to form a generative system. During a SG information on the player is only known at the source level and at the data level; we know its inputs and outputs, and can collect the information the player produces, but we do not know what happens within the system of the player. Researching how the other levels of the player can be analyzed and modeled would be an interesting research question.

REFERENCES

Caluwé, L. d., Hofstede, G., & Peters, V. (2008). *Why do Games Work? In Search of the Active Substance.* Deventer, The Netherlands: Kluwer.

Crobach, C. (2012). Developing a Formal Framework for Simulation Games. Delft, The Netherlands: TU Delft.

Crookall, D. (2010). Serious Games, Debriefing, and Simulation/Gaming as a Discipline. *Simulation Gaming, 41*(6), 898-920.

Duke, R. (1974). *Gaming: the futures language.* New York, NY: John Wiley & Sons, INC.

Duke, R., & Geurts, L. (2004). *Policy games for strategic management: Pathways into the Unknown.* Amsterdam, The Netherlands: Dutch University Press.

Faria, A., Hutchinson, D., Wellington, W., & Gold, S. (2009). Developments in Business Gaming: A Review of the Past 40 Years. *Simulation & Gaming, 40*(4), 464-487.

Harteveld, C. (2009). Making sense of studying games: using sensemaking as a perspective for game research. In G. Kin, & Y. Cai (Ed.), *Learn to game, game to learn: Proceedings of the 40th ISAGA Conference.* Singapore: National University of Singapore.

Kampen, N. v. (2006). *Stochastic processes in physics and chemistry.* Amsterdam, The Netherlands: Elsevier.

Klabbers, J. (2003). The gaming landscape: a taxonomy for classifying games and simulations. In M. Copier, & J. Raessens (Ed.), *Level Up Digital Games Research Conference* (pp. 54-67). Utrecht, the Netherlands: Utrecht University.

Klabbers, J. (2006). *The Magic Circle: Principles of Gaming & Simulation* (Vol. 1). (J. Spector, N. Seel, & K. Morgan, Eds.) Rotterdam, The Netherlands: Sense Publishers.

Klabbers, J. (2006). *The Magic Circle: Principles of Gaming & Simulation* (Vol. 1). (J. Spector, N. Seel, & K. Morgan, Eds.) Rotterdam, The Netherlands: Sense Publishers.

Klir, G. (1985). Complexity: Some general observations. *Systems Research, 2*(2), 131-140.

Kolb, A., & Kolb, D. (2005). Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education. *Academy of Managemetn Learning & Education, 4*(2), 193-212.

Kolb, D. (1984). *Experiential learning: Experience as the source of learning and development.* London, UK: Prentice Hall.

Nidumolu, S., Menon, N., & Zeigler, B. (1998). Object oriented business process modeling and simulation: a discrete event system specification framework. *Elsevier: Simulation Practice and Theory, 6*(6), 533-571.

Peters, V., Vissers, G., & Heijne, G. (1998). The Validity of Games. *Simulation Gaming, 29*(1), 20-30.

Traoré, M., & Muzy, A. (2006). Capturing the dual relationship between simulation models and their context. *Simulation Modelling Practice and Theory, 14*(2), 126-142.

Verbraeck, A., & Valentin, E. (2006). *Discrete modellen Deel II Discrete Simulatie.* Delft, The Netherlands: TU Delft.

Wenzler, I. (1997). Take five: gaming/simulation design process. In V. Bisters, *Simulation and Gaming for Sustainable Development* (pp. 251-259). Riga, Latvia: Environmental Publishers Vide.

Wymore, W. (1977). *Mathematical Theory of Systems Engineering: The Elements.* Huntington, NY: Krieger Publishing Company.

Zeigler, B. (1984). *Multifacetted Modelling and Discrete Event Simulation.* London, UK: Academic Press.

Zeigler, B., Prachofer, H., & Kim, T. (1976). *Theory of Modeling and Simulation.* San Diego, CA: Elsevier Science.

INTERVIEWS

Klabbers, J. (2012, October 24). Developing a formal framework for simulation games. (C. Crobach, Interviewer)

Meijer, S. (2012, October 23). Evaluation of the formal framework. (C. Crobach, Interviewer)

Warmelink, H. (2012, September 27). Evaluation of the Formal Framework. (C. Crobach, Interviewer)