Synchronising Wisdom and Implementation: A Formal ODD Approach to Expressing Insights on Bullying

Themis Dimitra Xanthopoulou

UiA University of Agder

Doctoral Dissertations at the University of Agder 441

Synchronising Wisdom and Implementation: A Formal ODD Approach to Expressing Insights on Bullying

Themis Dimitra Xanthopoulou

Synchronising Wisdom and Implementation: A Formal ODD Approach to Expressing Insights on Bullying

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* at
the Faculty of Engineering and Science, Specialisation in ICT, Multimedia and e-Learning

University of Agder
Faculty of Engineering and Science
2023

Doctoral Dissertations at the University of Agder 441
ISSN: 1504-9272
ISBN: 978-82-8427-158-3

Printed by 07 Media
Oslo

Acknowledgments

In 2018, I conducted a gratitude seminar as part of a bullying intervention plan. One of the participant assignments was to write a gratitude letter to a person the participant genuinely felt gratitude for. The challenge was not to write the letter but to read it aloud to the intended recipient. To exemplify how this should be done, I myself wrote a gratitude letter to someone I did not know but had helped a lot during a difficult time. Then, I read it aloud to her. To this day, I still remember the warm feelings that followed me hours after the recitation. There are a lot of people I want to thank. Without them, I may not have succeeded in completing this dissertation. However, expressing my gratitude in this section feels both very personal and completely impersonal. As a result, I choose to thank them personally with gratitude letters.

Themis Dimitra Xanthopoulou
Grimstad
January 2023

Abstract

The Social Simulation methodology, a mix of traditionally unassociated fields, utilises computer models to describe, understand, predict, and reflect on social phenomena. The model creation process typically requires the integration of knowledge insights from academic and non-academic knowledge holders.

To ensure model quality, different processes are established in an effort to verify the alignment of knowledge insights and their implementation in the simulation model by the modelling team. However, due to a lack of technical skills, knowledge holders, who may not fully understand the model code, often perform these verification checks indirectly, for example by evaluating conceptual model descriptions.

Initially motivated to create quality models of social conflict, this dissertation approaches the model quality challenge with a Domain Specific Modelling (DSM) approach. The objective was to develop a DSM tool using the Design Methodology, supplemented by a case study to provide first-hand experience with the quality challenge. Based on our project requirements, we selected University bullying as the case study subject. The Design Methodology included the problem exploration, the identification of a DSM solution, the selection of a domain and programming language for the DSM tool, the agile development of the domain language aspects with test models, and a final evaluation using the case study model.

The solution involved creating a DSM tool with a user-friendly interface that utilises a high-level domain language fit for conceptual model descriptions. Knowledge holders can use the tool to express their insights in the domain language. Model code is then automatically generated from these descriptions.

The ODD protocol is a widely used documentation method in the Social Simulation community. To build the DSM tool, we developed the concept of a Formal ODD - an ODD protocol with a formal representation of the computer model code and a requirement for complete descriptions. We proceeded to develop a Formal ODD language.

The DSM tool was built using the MPS platform, with the Formal ODD as the domain language and NetLogo as the programming language. The process of writing a model description using the Formal ODD language is similar to writing an ODD protocol but with added sections for increased model transparency. We recommend the adoption of the Formal ODD as it reduces model creation time, eliminates potential coding errors, provides accurate model descriptions for replication and transparency, and reduces the risk of insight misalignment. Our experience with the bullying case study exposed concept ambiguity as a potential disruption of model quality. Our own Formal ODD implementation assists in conceptual clarification.

The significance of our main contribution, the DSM tool, is a solution to the misalignment problem by avoiding the implementation of insights by the modelling team without requiring knowledge holders to acquire programming skills. The dissertation also contributes a new framework to understand the modelling stages. The framework was synthesised using the Object Management Group hierarchy and the new concepts of model physics and model context. Finally, we provide a base model for social simulation models related to exclusion and marginalisation.

Norwegian Summary

Sosiale simulering, en blanding av normalt adskilte felt, bruker datamodeller for å beskrive, forstå, forutsi og reflektere over sosiale fenomener. Modellene skapes ved å integrere kunnskap fra akademiske og ikke-akademiske kunnskapsbærere.

Modellkvaliteten sikres gjennom ulike prosesser av modelleringsteamet som skal verifisere samordningen av kunnskap og implementeringen av den i simuleringsmodellen. På grunn av manglende tekniske ferdigheter vil kunnskapsbærere noen ganger ikke forstå modellkoden fullt ut. Derfor utføres verifikasjonen ofte indirekte, for eksempel ved å evaluere konseptuelle modellbeskrivelser.

Denne avhandlingen takler kvalitetsutfordringer med domene-spesifikk modellering (DSM), og ble opprinnelig motivert av behovet for å lage kvalitetsmodeller for sosial konflikt. Det ble utviklet et DSM-verktøy med designmetoden, supplert med en case-studie om universitetsmobbing som gav førstehåndserfaring med kvalitetsutfordringene. Designmetodikken inkluderte problemutforskning, identifisering av en DSM-løsning, valg av domene og programmeringsspråk for DSM-verktøyet, smidig utvikling av domenespråket med testmodeller, og en sluttevaluering ved bruk av casestudiemodellen.

Løsningen innebar å lage et DSM-verktøy med et brukervennlig grensesnitt som bruker et domenespråk på høyt nivå som passer for konseptuelle modellbeskrivelser. Kunnskapsbærere kan bruke verktøyet til å beskrive sin kunnskap i domenespråket. Modellkode genereres deretter automatisk fra disse beskrivelsene.

ODD-protokollen er en mye brukt dokumentasjonsmetode i sosial simulering. For å bygge DSM-verktøyet utviklet vi konseptet for en formalisert ODD - en ODD-protokoll med en formell representasjon av modellen, inkludert et krav om fullstendige beskrivelser. Vi brukte dette konseptet for å utvikle et formelt ODD-domenespråk.

DSM-verktøyet ble bygget ved hjelp av MPS-plattformen, med Formal ODD som domenespråk og NetLogo som programmeringsspråk. Prosessen med å skrive en modellbeskrivelse ved å bruke det formelle ODD-språket til DSM-verktøyet vårt ligner på å skrive en uformell ODD-protokoll, men med ekstra seksjoner for økt modellgjennomsiktighet. Vi anbefaler å ta i bruk den formelle ODD siden den ikke bare reduserer modelleringstiden og eliminerer potensielle kodefeil, men også gir nøyaktige modellbeskrivelser som muliggjør replisering og åpenhet samt reduserer risikoen for feil. Vår erfaring med casestudien om mobbing avdekket begrepsuklarhet som en potensiell feilkilde av modellkvaliteten. Vår egen formelle ODD-implementasjon hjelper til med konseptuell avklaring.

Vårt hovedbidrag, DSM-verktøyet, er en løsning på verifiseringsproblemet ved å formelt beskrive kunnskap fra modelleringsteamet, uten at kunnskapsbærere må tilegne seg programmeringsferdigheter. Avhandlingen bidrar også med et nytt rammeverk for å forstå stadiene

i modelleringsprosessen. Rammeverket ble opprettet basert på Object Management Group sitt hierarki og de nye konseptene modellfysikk og modellkontekst. Som resultat av casestudiet gir vi en grunnleggende modell for sosiale simuleringsmodeller knyttet til eksklusjon og marginalisering.

Publications

Included Works

The following papers are included as a part of this thesis. They have been published or submitted in peer-reviewed conference proceedings and journals. The included versions in this dissertation may differ in formatting compared to the original published versions.

Paper A Themis D. Xanthopoulou, Andreas Prinz, and F. LeRon Shults. Generating executable code from high-level social or socio-ecological model descriptions. In Pau Fonseca i Casas, Maria-Ribera Sancho, and Edel Sherratt, editors, System Analysis and Modeling. Languages, Methods, and Tools for Industry 4.0, pages 150–162. Springer International Publishing, 2019

Paper B Themis D. Xanthopoulou, Ivan Puga-Gonzalez, F. LeRon Shults and Andreas Prinz, Modeling Marginalization: Emergence, Social Physics, and Social Ethics of Bullying, In 2020 Spring Simulation Conference (SpringSim), Fairfax, VA, USA, 2020 pp. 1-12. doi: 10.22360/SpringSim.2020.HSAA.005

Paper C Themis Dimitra Xanthopoulou, Andreas Prinz, and F. LeRon Shults. The problem with bullying: Lessons learned from modelling marginalization with diverse stakeholders. In Marcin Czupryna and Bogumil Kamiński, editors, Advances in Social Simulation, Springer Proceedings in Complexity,
pages 289–300. 2022, Springer Nature

Paper D Themis D. Xanthopoulou., Andreas Prinz (2022). Informal Model vs Formal Model: Human Experience vs Conceptualized Model Descriptions [Submitted Paper]

Paper E Themis D. Xanthopoulou, Andreas Prinz, Ivan Puga-Gonzalez, F. LeRon Shults (2022). Formal ODD: Exposing Model Physics and Model Context [Submitted Paper]

Additional Contributions

In addition to the papers included in this thesis, the author has contributed to the following papers during the PhD duration.

Paper F Andreas Prinz, Themis D. Xanthopoulou, Terje Gjøsæter, and Birger Møller- Pedersen. On abstraction in the omg hierarchy: Systems, models, and descriptions. In Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS '22, page 322–330, New York, NY, USA, 2022. Association for Computing Machinery.

Paper G Themis D. Xanthopoulou, Andreas Prinz, F. LeRon Shults, Ivan Puga-Gonzalez, and Ingrid Lund. Social Simulation of University Bullying. 2019. Poster presented at the Antibullying Conference 2019, Dublin, Ireland

Paper I Themis D. Xanthopoulou, Saga Pardede, and Andreas Prinz. Bully-proofing a learning environment. 2021. Paper presented at the Antibullying Conference 2021, Stockholm, Sweden.

Contents

1 Overview **1**

1.1 Problem Statement 2

1.2 Approach 4

1.2.1 The Case Study 5

1.2.2 Research questions 6

1.3 Answers and Contributions 7

1.3.1 Answers to the Research Questions 8

1.3.2 Contributions 10

2 Social Simulation **13**

3 Planned Methodology **17**

3.1 Building the bullying model 18

3.1.1 Collaborations 18

3.1.2 Model development 20

3.1.3 Model Development Steps 23

3.2 Domain Specific Modelling 25

4 Exploring the Bullying case study **27**

4.1 Specification of the case study (Step 1) 27

4.2 Establishing collaborations (Step 2) 29

4.3 Consultations and Stakeholder Meetings (Step 3) 29

4.3.1 Knowledge Holder Consultations 29

4.3.2 Stakeholder Meetings 30

4.4 Modelling Choices and First Version (Steps 4-7) 31

4.5 Feedback from the Antibullying forum (Step 8) 33

4.6 Reflection results and Informal Interviews (Step 9) 34

4.7 Is there a consistent understanding of the term bullying in current interventions-Implications (Step 10) 39

4.8 The neurological basis of conceptualisation and miscommunication of knowledge holder insights 41

5 Developing the DSM tool **47**

5.1 Applied Methods 47

5.2 Problem exploration (Step A) 49

5.2.1 The OMG Hierarchy . . . 50
5.2.2 Model Context and Model Physics . . . 51
5.2.3 Improved understanding of the model stages . . . 52
5.2.4 Challenges and Errors . . . 54
5.2.5 Existing Methods to Approach the Stated Problem . . . 56
5.3 Proposed Solution (Step B) . . . 58
5.4 Design Choices (Steps C to G) . . . 60
5.5 Feedback for the tool (Steps H and 11) . . . 61
5.6 Solution Summary . . . 62
5.7 Dissemination and Use (Step I) . . . 63

6 Discussion 65
6.1 Precision vs Imprecision . . . 65
6.2 ODD2NetLogo, Knowledge Representation and Knowledge Engineering . . . 67
6.3 Alternatives . . . 68
6.4 Limitations and Future Steps . . . 70

7 Summary 73

Bibliography 75

Appended Papers 87

A Generating Executable Code from High-Level Social ... 87

B Modeling Marginalization: Emergence, Social Physics, and ... 103

C The Problem with Bullying: ... 117

D Informal Model vs Formal Model: ... 131

E Formal ODD: Exposing Model Physics and Model Context 151

F Social Simulation of University Bullying 177

Documentation of ODD2NetLogo 181

Generation Rules 181
a Concepts from the Folder "4FutureUse" . . . 181
b Concepts from the Folder "Actions and Procedures" . . . 181
c Concepts from the Folder "Appearence" . . . 190
d Concepts from the Folder "EntitiesAndAttributes" . . . 191
e Concepts from the Folder "Experiments" . . . 191
f Concepts from the Folder "Experiments" . . . 194
f.1 Concepts from the Subfolder "Condition" . . . 194
f.2 Concepts from the pool of the folder "Experiments" . . . 196
g Concepts from the Folder "Initialization" . . . 201

h Concepts from the pool of concepts 204

List of Figures

1.1 A brief description of the Modelling Process 3
1.2 Overview of Domain-Specific Modelling inspired by [1] 4
1.3 Model vs Model and Thesis Structure 7

3.1 Steps in the Design Approach inspired by [2] 17
3.2 Model Stages and Flow of processes in a Social Simulation project 21
3.3 Planned Methodologies with Detailed Steps 25

4.1 The frontpage of UiA website 28
4.2 The Sphynx cat and the Oncilla 35
4.3 Two very different types of cranes 36
4.4 Bullying at Different Levels of Abstraction (Retrieved from Paper C and reproduced with permission from Springer Nature) 37
4.5 A Model of Modelling 41
4.6 Data collection of human experiences: from the conception to the social simulation model drawn from Paper D 42
4.7 Model of Conscious Perception 43
4.8 Sources of errors in conceptualisation and communication of human experiences 44

5.1 Applied Methodological Steps: Red indicates unfinished planned steps, while green represents unplanned but realised steps 47
5.2 A knowledge map for Figures 4.5, 4.6, and 3.2 48
5.3 Model transitions during model creation (adapted from Paper E) 52
5.4 Languages for different Model Descriptions, extracted from Paper E 53
5.5 Proposed Solution in the DSM framework adapted from (Paper E) 58

7.1 Summary of the presented work 74

List of Tables

4.1 Input grouped in categories retrieved from Paper C and reproduced with permission from Springer Nature . . . 34
4.2 Evaluation of the KIVA program . . . 40

Chapter 1

Overview

The dissertation you are reading builds on the work conducted for the project entitled "Computer Tools for Modelling Social Conflict" (CTMSC). Four members with diverse backgrounds made up the project team with the aim to join forces from our corresponding fields to develop *Computer Tools* to support *Models of Social Conflict*. The term "Models of Social Conflict" refers to the Social Simulation Models built with the Agent-Based Modelling method (ABM). Within the project restrictions was the selection of the Domain Specific Modelling (DSM) approach. Note that the project goal was not to develop conflict models but to support the modelling process by developing computer tools. Naturally, the thesis theme is drawn from the project context and inherits the project restrictions. Throughout this PhD dissertation, I am exchanging pronouns from "we" to indicate the group work to "I" to show the dissertation work.

Models of social conflict can be beneficial for conflict resolution. Conflict is present in many aspects of our lives and has dominated the news even in the relatively short period of the project work. From disputes over election results to actual armed conflict, people fight each other, and a lot of times, we are none the wiser to stop it. Conceptual social models result from processes of knowledge acquisition [3]. One of their roles is to help us structure and describe the collected knowledge over conflict. Computational social models represent conceptual models in a computer environment, written in a programming language [3]. Social simulation models built with the Agent-Based Framework are a category of computational social models. Computers allow us to experiment with knowledge and connect with reality. Conflict models can be both conceptual (such as [4]) and computational (such as [5]). In both cases, advice drawn from the models helps us understand conflict and hopefully resolve it.

In this work, we chose to focus on model quality issues of social simulation models. The motivation is to increase model quality because this will lead to better advice and conflict management. The dissertation is structured into two main parts: the core thesis text and the supplementary materials, which consist of both published and submitted research papers, as well as part of the documentation. It is possible to read the thesis without the publications, but they provide more details. The goal of this Overview Chapter is to briefly present the thesis theme in the problem statement section (Section 1.1), the approach (see Section 1.2), the reasons for selecting the bullying case study in Section 1.2.1, the research questions in Section 1.2.2, brief answers to the research questions in Section 1.3.1, and the scientific contributions attached to the thesis work in Section 1.3.2. The Social Simulation Chapter (Chapter 2) provides background information about Social Simulation and the general methodological benefits of computational social models. In the Planned Methodology Chapter (Chapter 3), I go deeper into the approach and analyse the planned methods with detailed steps. The results are distributed in two Chapters: Chapter 4 and Chapter 5. Chapter 5 contains a detailed problem exploration in Section 5.2. To maintain a good flow in the text, state-of-the-art methods are presented in various chapters, with a comprehensive review of relevant previous work for the updated problem statement in Section 5.2.5. In Chapter 6, I discuss the dipole of precision-imprecision, the relationship of Knowledge Engineering with the project work, the project limitations and potential alternative routes. The thesis is concluded with a summary in Chapter 7.

1.1 Problem Statement

Creating Social Simulation models requires the collaboration of researchers from different scientific fields. Often, we need subject experts to convey specific knowledge and a modelling team to build the simulation model. If you have ever been part of such a collaboration, you will recognise the communication difficulties. Sometimes, instead of subject experts, we rely on knowledge from regular people with first-hand experience. For this reason, I will use the general term "knowledge holder" to indicate the person who owns and provides accumulated knowledge over a domain. In addition, the term "insights" signifies the chunks of knowledge communicated to the model creation team during the data collection process or via communication with collaborators.

To better understand the problem we addressed, Figure 1.1 shows a brief overview of the stages and transitions in a Social Simulation project. I am describing the process starting with the creation of a new conceptual model (Conceptual Model Stage in Figure 1.1). In this stage, the modelling team decides whether to keep or discard the knowledge insights provided by the knowledge holders. Then, the conceptual model is written in a programming language and becomes a computer model (Computer Model Stage in Figure 1.1). The computer model is explored with simulations, and a new stage of knowledge is achieved when the simulation outcomes are analysed. In this work, the term "Simulation Model Stage" in Figure 1.1 refers to this new stage of knowledge. Conclusions from the simulation model can be used as advice to address social conflict. The model stages of Figure 1.1 can be found in published frameworks of the modelling process (such as [6]). Whereas transitioning from the computer model to the simulation model is organic (Transition 2 in Figure 1.1), transitioning from the conceptual model to the computer model (Transition 1 in Figure 1.1) is challenging. There are many reasons for the difficulty in Transition 1, but we focused on how the knowledge insights are transformed from descriptions and words to programming code.

In the field of Computer Science, particularly Artificial Intelligence, Knowledge Representation is a significant subject devoted to studying how to encapsulate knowledge in a machine-readable language to enable computations [7, 8]. Two components are necessary for Knowledge Representation: ontology, which denotes the entities and their interrelationships, and logic, defined in Sowa's book as the "formal structure and rules of inference" [7] . The meanings we infer in natural language as humans are translated into an ontology to enable the

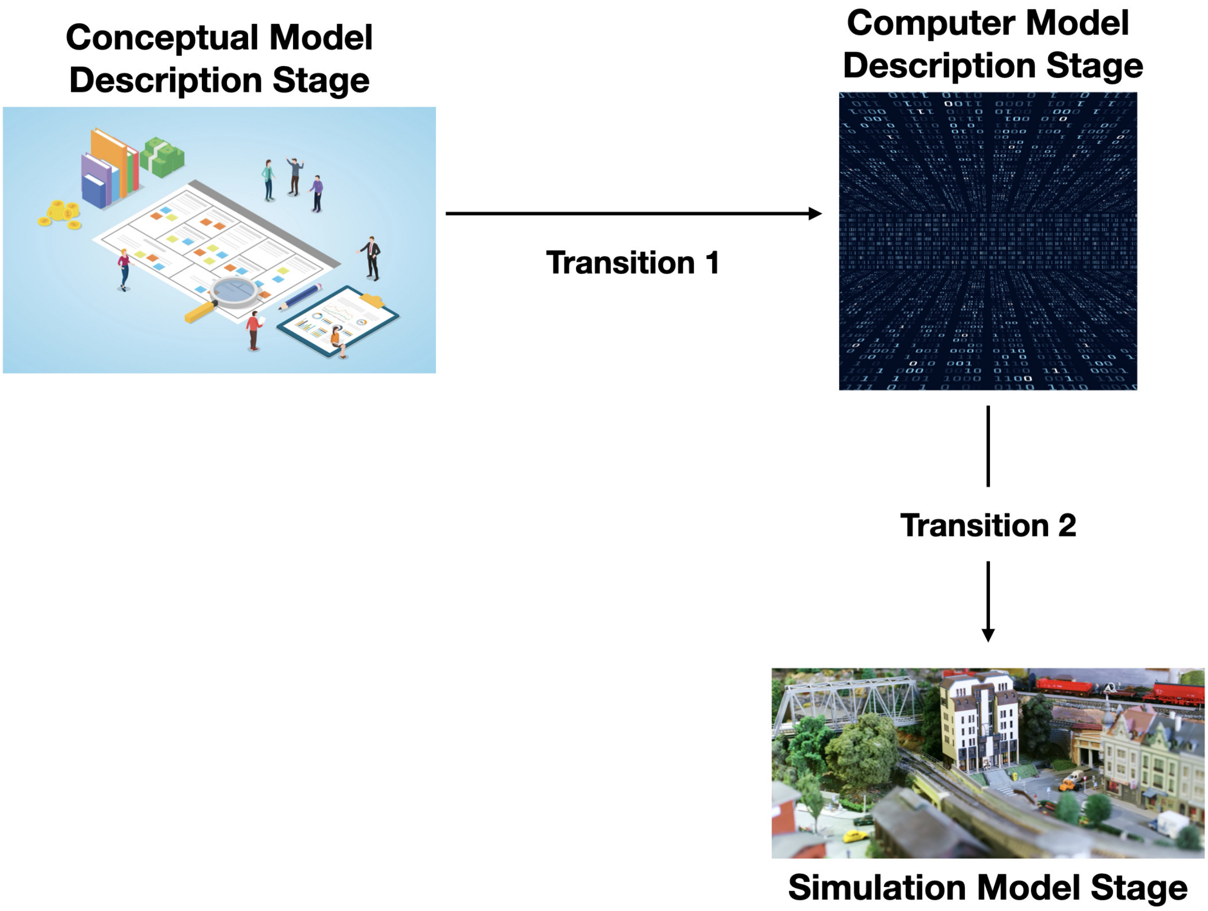


Figure 1.1: A brief description of the Modelling Process

use of knowledge by computers [9]. Though the process begins with the knowledge insight provided by the knowledge holders [8], it continues with the work of knowledge engineers. These experts, which in our case form the modelling team, are challenged by the complex procedures required for effective knowledge representation [7]. The construction of ontology and its articulation in a Knowledge Representation language necessitates the introduction of assumptions. This fact brings forth questions of fidelity between the initial insight and the end product—in this case, the computer model description [8]. Within the social simulation community, this challenge has been discussed within the context of operationalisation, which is the process of quantifying qualitative terms.

Simulations function as tools to create new knowledge derived from computer model descriptions [7]. Given that this new knowledge is fundamentally based on the computer model description, it becomes crucial to assure the quality of knowledge representation within this description. Contrarily, translating the newly acquired knowledge from simulation outcomes back into natural language is a relatively straightforward process. This ease of translation is largely due to the modelling team's familiarity with the engineered transitions in the knowledge representation process.

> **Problem statement: How do we make sure that the insights provided by the knowledge holders are accurately represented in the social simulation models?**

Based on this problem formulation, the objective became the design of a computer tool that would assist the model creation process in Transition 1, especially concerning how knowledge-holder insights are implemented in the operationalisation process. A correct match between knowledge holder insights and implementations in the different modelling stages would ensure part of the model quality.

1.2 Approach

The CTMSC project required the use of Domain-Specific Modelling to address the problem at hand. The information in this paragraph about DSM is drawn from the book by Kelly and Tolvanen [1]. DSM is a methodology that can be applied to various domains beyond computer science, where software solutions are used to solve problems. It is handy for domain experts who lack programming expertise. As Kelly and Tolvanen describe in their book, the main idea behind DSM, as illustrated in Figure 1.2, is the creation of a DSM tool that acts as an intermediary between the description of the solution in a language familiar to the domain experts and the solution expressed in the programming language. The DSM tool automatically generates the solution description from the "domain language", which is a high-level language, to the programming language (referred to as "automatic code generation" in Figure 1.2). It also translates the current solution from the programming language to the domain language through an abstraction process (as shown in Figure 1.2). This enables domain experts with limited programming expertise to create, read, evaluate, and modify solutions by working with the description in the domain language. At the same time, the DSM tool handles the rest of the work.

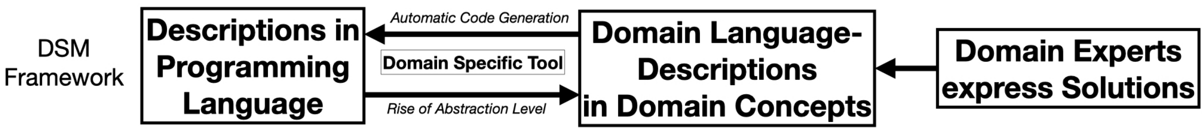


Figure 1.2: Overview of Domain-Specific Modelling inspired by [1]

The methodology we incorporated to understand and address social conflict included the use of a case study. This case study would provide us with firsthand experience modelling social conflict, allowing us to fully explore the problem and choose an appropriate solution. The second function of the case study would be to evaluate the designed solutions, which is one of the steps in the design methodology according to Peffers et al. [2]. Therefore, selecting an appropriate case study was crucial to the success of our project.

To ensure that our case study was both relevant to our project and informative about the complexities of social conflict, we established three key requirements. First, the case study had to be directly related to social conflict. Second, it had to offer ample opportunities for interaction with knowledge holders, which was an essential aspect of our approach to model quality. Finally, the case study had to involve a type of social conflict that would benefit from the social simulation methodology. By carefully considering these requirements, we were to select a case study that would be both useful and meaningful.

1.2.1 The Case Study

We selected Bullying as our case study. The case study of bullying satisfied all our three requirements:

- **Requirement 1.** Bullying is a type of social conflict with research interest. To begin with, it takes place in all aspects of our lives where we have direct or indirect interactions with others. For instance, we find bullying dynamics in the family environment, school environments, workspace, and online interactions (aka cyberbullying) [10–13]. Moreover, bullying has been linked with mental health issues, and violence [14, 15]. In addition to health issues, bullying affects performance. For example, bullied students are more likely to withdraw from their education [16], and employees subject to workspace bullying have lower productivity [17]. Overall, everyone can be potentially exposed to bullying, and, at the same time, research suggests that the severity of consequences can be very high.

- **Requirement 2.** The proximity to knowledge holders and stakeholders at UiA fulfilled the second criterion for the selection of bullying as our case study. At the start of the project, we had a lot of indications that we could easily attract collaborators to work on a bullying case study. In 2017, Ingrid Lund (Professor at UiA) surveyed the bullying situation at the University [18]. She discovered that almost 10% of University students were being bullied. More than a decade before this survey, Norwegian legislation had marked bullying as illegal and had requested institutions and companies to take active steps in preventing and addressing it [19]. As a result, the survey output attracted the attention of UiA's then-rector Karl-Heinz Frank Reichert. Frank then formed a workgroup to address bullying and announced that there would be zero tolerance towards bullying. [20]. The announcement occurred at the same time we started the CTMSC project.

- **Requirement 3.** The social simulation methodology may solve the current research gap regarding bullying interventions. Even if interventions are necessary to prevent bullying, they remain an enigma. That is not to say that suggested interventions do not exist. The European Union has collected several interventions [21] and has posted them on its official website [22] alongside an evaluation for each program. The main issue is that interventions do not consistently have positive outcomes [23–25]. At the same time, bullying is considered a very complex phenomenon [26]. Multiple fields, such as psychology [27], social sciences [28], education studies [29], etc., have contributed to bullying research. Social simulation can integrate all perspectives using the ABM method and thus address the complexity. Consequently, a social simulation model with an explanatory purpose could identify possible reasons why things have worked out well on some occasions and failed on others.

1.2.2 Research questions

Some of the research questions were predefined from the beginning of the project while others emerged as a result of the research progression. Overall, they follow two themes. The first theme directs the case study and the DSM tool design towards the improvement of model quality.

To gain a perspective on the limitations regarding model quality, A0 reflects on the inherent quality issues in the insights provided by knowledge holders. A1 and A2 are directed towards the improvement of model quality in relation to the representation of knowledge holder insights.

A0. What are the inherent limitations in forming accurate knowledge holder insights and retaining their essence during the communication phase with the data collection team?

A1. How can we model bullying while retaining the insights of knowledge holders throughout the modelling process shown in Figure 1.1?

A2. How can we gain a better understanding of the transitions of Figure 1.1 to improve the formulation of the problem and find a suitable solution? What function do we need in a DSM tool to support model quality during Transition 1 of Figure 1.1, with a specific focus on the implementation of knowledge holder insights?

The second theme guides the project's development phase by raising several questions related to the creation of the social simulation model and the DSM tool. Important components of the modelling process include the conceptual model and Transition 1 as defined in Figure 1.1. It was also essential to consider expected model outcomes, particularly concerning interventions, as this aligns with the research gap we are interested in. Parts of the DSM development process are the selection of the DSM function, the domain language and the programming language, as outlined in the DSM framework in Figure 1.2. The resulting questions, in accordance with the second theme, are:

B1. What is the ontology of bullying based on the knowledge holder insights? Is there a consistent understanding of the term bullying in current interventions?

B2. How can we operationalise bullying?

B3. How do we model interventions?

B4. What type of interventions are successful for bullying?

B5. What do we require from the DSM language? Are there available languages that meet our requirements?

B6. How do we use an existing language to serve as a DSM tool?

B7. What programming language should we choose?

1.3 Answers and Contributions

This Section provides brief answers to the research questions and the main contributions. Figure 1.3, which is based on Figure 1.1, illustrates the connection with the problem statement and incorporates the research outcomes. It serves as a guide to understand the relationship between the resulting publications and the thesis concepts. The results are presented in a logical sequence that resembles the chronological sequence we followed and facilitates understanding.

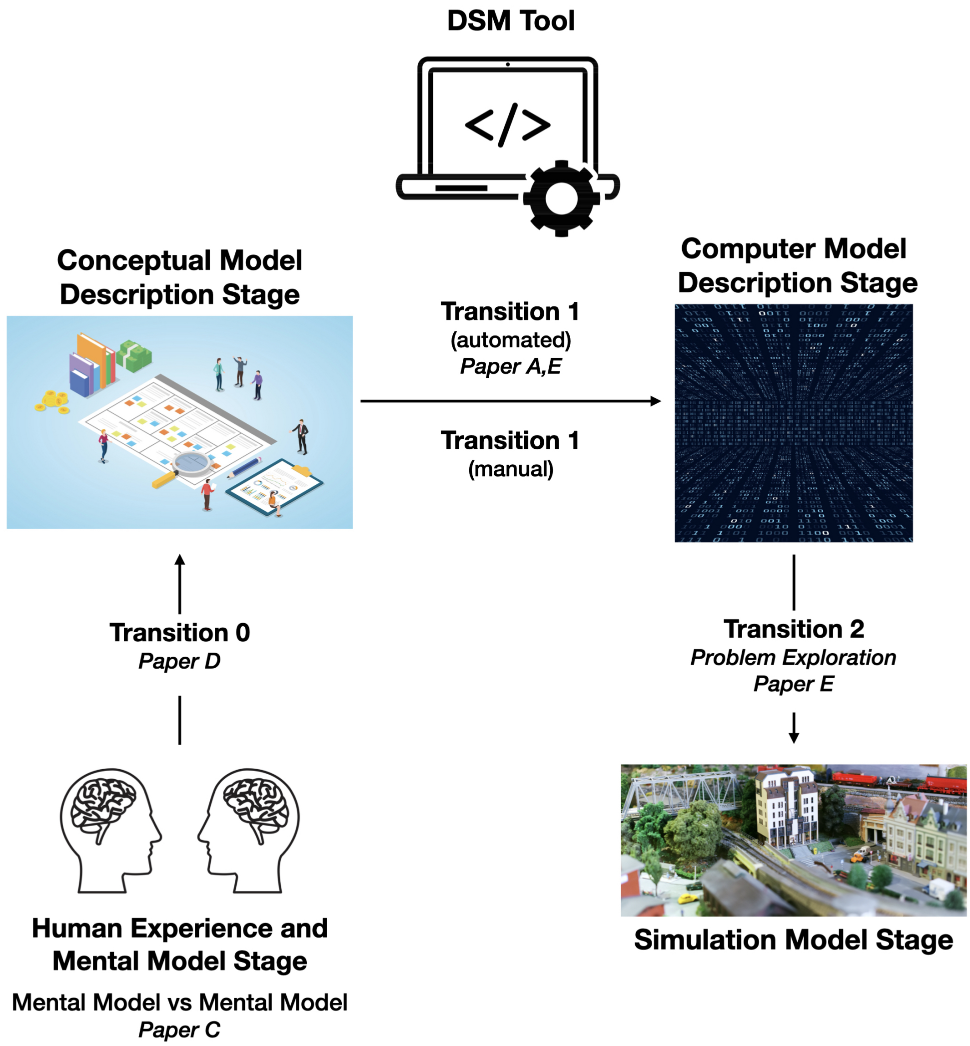


Figure 1.3: Model vs Model and Thesis Structure

1.3.1 Answers to the Research Questions

B2. We operationalised bullying using a recent definition emphasising inclusion-exclusion dynamics. From this perspective, a sense of being bullied arises when interactions perceived as rejected or negative form the majority of a student's total interactions. A full analysis of the operationalisation process is published in Paper B.

B1. The term bullying is ambiguous, and its ambiguity is not widely acknowledged. In general, bullying is considered an abstract term of action. First, academics disagree on bullying definitions and display variance in the interpretation of bullying. In addition, academic and non-academic knowledge holders do not operate with the same levels of abstraction. The former use definition, and the latter use descriptions of actions. Overall, they disagree on what constitutes bullying and there is no shared ontology of bullying. More details are found in Paper C. Upon investigating the available information on the Kiva intervention program, it's evident that the program employs a widely accepted definition of bullying, supplemented with examples, which nevertheless leaves ample room for interpretation. Given the ambiguity surrounding bullying, it's plausible that different stakeholders might implement the intervention measures under varying conditions, leading to inconsistent results. This analysis is elaborated upon in Section 4.7.

A1. We intended to build an explanatory bullying model and later extend it to include intervention strategies. Considering the ambiguity of bullying, insights from different knowledge holders cannot mix in a conceptual model because we must first clarify whether they have shared opinion regarding bullying. But even if this obstacle is overcome, there is a risk of misuse of the model outcomes under a different interpretation of bullying. For all these reasons, we propose the use of more concrete concepts of actions to avoid the compromise of model quality and later misuse of model conclusions. The full argumentation with details is published in Paper C.

B3/B4. For reasons presented in A1 and due to time restrictions implying that we could not explore the different versions of ontology for bullying, we decided not to proceed with the integration of interventions. Thus, we did not answer B3. We have a tentative response to B4 based on the ethical reflections allowed by our first model version (presented in Paper B). The normative character of bullying implies that we should trace our actions back to our values and reconsider the openness and acceptance they express. We additionally presented a framework to bully-proof a digital classroom in Paper I, which is cited in the Additional Contributions.

A0. Our brain's neural network architecture forms the foundation of our conceptual system, with neuron connections mapping out the associations we create. Perception, an integral part of understanding conceptualisation, is formed by exteroceptive, interoceptive, and proprioceptive signals. Only signals that exceed an intensity threshold, known as the absolute threshold, can reach our conscious perception. Further refinement occurs through perceptual filters and predictive coding, a process wherein previous experiences contribute to the construction of perception. Concepts - dynamic, context-dependent summaries of similar experiences - drive our predictive perception and inform our options for actions. But even if we considered conscious perception to be unbiased, our conceptualisation process is not flawless; biases and erroneous associations of stimuli can result in skewed mental models. The recall of experiences can be compromised by various factors such as conceptual changes, incomplete memory retrieval, and denial. Variations in the understanding of the same entities, as often occurs in different cultures, along with word ambiguities (as seen with terms like 'bullying') and mislabeling of our own experiences, can impede effective communication of insights. This underlines the importance of clarity and precision. Depending on our intended use of the knowledge insights, the inaccuracies in their expression can create a misrepresentation of the original experience in the social simulation model. Furthermore, during data collection, recollections of experiences can be misinterpreted, adding to the potential for conceptual model description inaccuracies. The detailed analysis is found in Paper D.

A2. Beyond the typical transitions involved in the model creation process, it's vital to account for the transformation of the knowledge insight from the mental model of the knowledge holder to the communicated, and subsequently written, representation of the same insight in the conceptual model description (Transition 0). The modelling process typically proceeds with the transformation of model descriptions rather than the models themselves. As the modelling process unfolds, these descriptions become more formal, functioning like a funnel that narrows down the range of potentially prescribed models. A crucial part of the model is the "model physics", which refers to the cumulative mechanisms that advance the simulation from one snapshot to another. One of the purposes of the social simulation modelling process is to thoroughly capture these model physics. For more analysis please refer to Section 5.2. Among the reasons for the potential mismatch between insights and their implementations is the potential difficulty of knowledge holders to read model descriptions in programming languages and the technical errors in Transition 1. We designed a DSM tool that automatically generates the computer model description from a domain language typically used for model documentation (Automation of Transition 1 in Figure 1.3). This automation allows knowledge holders who can write conceptual model descriptions to create computer model descriptions without intermediate interpretations. At the same time, the conceptual model description acts as model documentation because it is synched with the computer model description, enabling knowledge holders to read the implementation of insights. Consequently, the design allows knowledge holders to create and modify models. Based on the case study experience, we designed the DSM's user interface to accommodate further conceptual clarification. The process of building the DSM tool, including the answers to B5, B6 and B7, is presented in Paper A, and the resulting tool is shown in Paper E.

B5. We needed a domain language suitable for social simulation models' descriptions. We selected the ODD (Overview, Design concepts, and Details) protocol as the DSM's tool domain language because it is widely used to document modelling choices in ABMs. The DSM's user interface is structured as an ODD protocol. As a result, the experience of describing models in the DSM tool resembles the experience of writing an ODD protocol. The Social Simulation community's familiarity with ODD decreases our DSM tool's learning curve. Moreover, the preoccupation with and constant updates of the ODD protocol imply a state-of-the-art capacity to describe models and meet the community's needs.

B7. We selected the NetLogo language as the programming language of the DSM tool. NetLogo itself is designed to code social simulation models. As a result, the transition rules from the ODD language to NetLogo were easier. In addition, NetLogo offers a simulation environment and the possibility of designing simulation experiments in bulk. The NetLogo platform is user-friendly, allowing knowledge experts and stakeholders to experiment with simulations easily.

B6. The ODD protocol is organised into sections containing specific questions. To formalise this protocol, we have integrated its existing structure into the DSM tool, requesting the user to answer the same set of questions but with tighter constraints on possible responses. To facilitate the automatic transition from user specifications to NetLogo code, we have devised transformation rules inspired by NetLogo, ODD, and other concepts used in the Agent-Based Framework. To ensure user responses adhere to the appropriate format, the editor incorporates constraints that either signal errors or prohibit random input. Additionally, the editor provides a list of potential answers to further guide users towards providing structured specifications.

1.3.2 Contributions

The thesis contributed to the theoretical understanding of the modelling process, the bullying phenomenon, and solutions regarding model quality issues.

1. We provide *an extended model of the modelling process* with two extra features: the Mental Model stage and Transition 0. These elements, depicted in Figure 1.3, represent how our brain conceptualises experiences (such as conflict experiences) and how we communicate them. In Paper D, we discuss and analyse the Model stages, Transition 0, and the potential errors that may arise from these additions to the model. By incorporating this additional stage and process, we aim to enhance the accuracy and comprehensiveness of our models.

2. We contributed to the field of bullying research with *a new understanding of the concept of bullying*. We achieved this by examining the concept of bullying and its inherent ambiguity, suggesting that it matches a concept of moral judgement, proposing alternative concepts to consider, and explaining the inconsistent results of interventions. Our complete analysis is presented in Paper C. Through our contribution, we hope to deepen

the understanding of what has been labelled as bullying and inform the development of more effective interventions in the future.

3. The *MARG model*, the product from working with the bullying case study, can act as a base to create future social simulation models for marginalisation. Even though we have discarded bullying as a modelling subject, efforts aligned with the antibullying movement could still be supported with the MARG model.

4. To fully explore the problem, we contributed with a state-of-the-art *framework for models* presented in Paper E. The analysis utilises the Object Management Group hierarchy framework and introduces two new concepts: the concepts of "model physics" and "model context". We also emphasised the difference between model and model description and captured it in the labels of the stages of Figure 1.3.

5. To design our DSM tool, we developed the *concept of a Formal ODD*, which bridges the DSM approach with the ODD protocol. Even though we used the Formal ODD to implement our solution, we believe it can be used outside the context of our problem statement. The benefits of a Formal ODD approach are presented in Paper E.

6. We implemented the *Formal ODD language* as part of our DSM tool. The Formal ODD language resembles and extends the ODD protocol. More details can be found in Paper E.

7. The developed *DSM tool* automatically generates model descriptions in programming languages from model descriptions written in the Formal ODD language. As a result, it offers a quick way to create and modify ABM models. Moreover, it includes a reliable description to understand model components, including the implementation of knowledge holder insights. Finally, it automatically provides model documentation. The methodology we used is presented in A and the tool itself in Paper E.

Chapter 2

Social Simulation

This chapter explores the importance of computational social models in Social Sciences and provides background information on the Social Simulation methodology and the Agent-Based method.

The use of models in social sciences is part of the social science research methodology. The social science methodology contains a series of steps starting from the research design and leading to data collection and data analysis [30]. Experiments such as the famous Milgram or the Stanford prison experiment are used as a source of data, but they are not always part of the methodology. Both the research design and data analysis phases include theory creation processes. Social theories are abstractions and precursors of conceptual social models [31]. While some researchers consider social theories separate from social conceptual models [30], we can view social theories as a type of social model. If we also think conceptual frameworks as social models, then the role of social models is well-established in the social sciences. Social models are descriptive by nature and informal for the most part. One of their functions is to help organise knowledge around a subject [32].

Examples of social models include a model that focuses on the emotional needs of victims and perpetrators [33], a model for the diffusion of bias in groups [34], a model for the spread of news and beliefs [35], and a model for learning in organisations [36]. Most of these models have resulted in suggestions for real-life interventions. But what is the status of formal models in social sciences?

It is not easy to assess the position of formal models in social sciences, but it would be fair to say that they are not so well established. Formality [1] implies that the semantics of descriptions are explicitly indicated. In other words, it is not possible to misinterpret a formal description. Two examples of formal descriptions would be mathematical formulas and computer programs expressed in a general-purpose programming language. Formal models in social science that are only used for data analysis, such as equation-based models, are not included in my consideration. These statistical and mathematical models, mentioned in [37], are not relevant in this context. Some indications for the limited use of formal models in social sciences arise from the fact that performing a literature search with keywords "formal" and "models" does not yield many results in popular social science journals. That being said, it was still possible to trace formal models in a social research handbook [38].

[1] Throughout the thesis I use the term formal to denote primarily formality in description and not formality in semantics.

The system dynamics method introduced formal models to the field of social science in the 1960s, when Jay W. Forrester used engineering principles to analyse business contexts [39]. Later, he expanded the use of system dynamics to urban policy and other fields, such as ecology, psychology, politics, and education [39]. The emergence of computational social science in the final quarter of the 20th century strengthened the position of formal models in social science, including the system dynamics and agent-based modelling (ABM) methodologies [40]. The term "social simulation" is often used as an umbrella term for the various methods found in computational social science.

Social simulations can be considered accelerated thought experiments. While some researchers have made the link of social simulations to thought experiments [41–43], this view is not widely held. As humans, we have limited memory and an inability to process complex, non-linear systems, which may lead to mental fatigue and incomplete consideration of all possible outcomes. On the other hand, social simulation allows us to study complex social systems in a controlled and systematic way, taking into account many variables and factors. Additionally, social simulations can generate predictions about how a social system might behave under different conditions, allowing researchers to test hypotheses and explore the potential outcomes of different scenarios. Social simulations provide an enhanced tool for analysing and understanding social systems in the social sciences compared to regular cognitive analysis.

We build social simulation models for many different reasons and with different purposes in mind [44, 45]. One big category of modelling purposes is to understand a particular social reality of interest. Social simulation models have enabled the rise of a new paradigm with the potential to improve our comprehension of social reality: generative social science [46]. As Joshua Epstein mentions in [46], the motto behind this paradigm is that "If you didn't grow it, you didn't explain its emergence.". In other words, if a simulation outcome matches reality, how we arrived at this result (our model ingredients and initialisation settings) may offer one possible explanation for the reality we are trying to understand.

In addition, to their explanatory capacity, social simulation models can be used to predict parts of our social world [44]. Predictions are often used to inform decision-making. In other words, they have real-life implications. Whenever we create simulation models with an expected impact on policy decisions or other implications in the non-academic world, we must be more rigorous with our modelling processes.

The most prominent methods are System Dynamics, Microsimulation, and Agent-Based Modelling [47]. Of all the Social Simulation methods, we are particularly interested in ABMs. ABM has been evaluated as less efficient and more demanding in its development but allows to "model active objects" [48]. This means it permits more options about how people interact and space for irrational or random behaviour.

A simple description of ABM is a "micro-simulation based on individually (inter)acting sub-models" (by inter(acting) sub-modes the author refers to the agents) [49]. Using the ABM framework, we conceptualise the world with entities, an environment, states, and behaviours. Each ABM has specified entities and one or more environments. We start by assigning behaviours to the entities and the environment. These behaviours can vary based on the entity type or the environment type. Next, we specify their states. As the simulation progresses, the entities interact with each other and the environment, thus changing their states. Finally, some parts of the behaviours could be random, accounting for unknown factors. The bottom-up character of ABM, especially compared to the top-down character of system dynamics, adds to its capacity to portray emerging phenomena. As a result, ABM becomes a powerful tool for understanding and portraying complex social realities.

ABMs hold the capacity to simulate interdisciplinary modelling subjects. For instance, there are socioecological models [50] with subcategories such as socio-hydrological models [51]. Another example is socio-economic models [52]. The integration is done by incorporating entities and parameters that are not purely social and by using multi-disciplinary theories to code the behaviours. As a result, ABMs have become a robust methodology for interdisciplinary research.

On a final note, there are diverse views on what increases the quality of an ABM model. The diversity is partly explained because opinions on quality depend on one's philosophical perspective of reality [53, 54]. For example, Ahrweiler and Gilbert, in their chapter [53], write that under what they call the "standard view", quality is defined in terms of the model verification and validation processes, while under the constructionist view, an evaluation is not possible since both models and observations are considered constructions. In connection with philosophical views, researchers emphasise different aspects when they refer to model quality. For instance, Barreteau et al. [55] consider model quality to depend on the realistic nature of results and model efficiency to depend on the relationship between model complexity and time needed for model exploration. Ahrweiler and Gilbert in [53] recommend the "User Community View" under which quality is found in making the model. More specifically, they consider quality to be in the interactions between stakeholders and the model creation team and the model's usefulness to the community. Another concept that has been linked to model quality is model correctness. Correctness has been given various definitions; an example is the model capability of "faithfully transmitting the causes to the outcomes" [56]. In this work, we focused on one aspect of model quality and model correctness, the alignment of knowledge-holder insights with their implementations in the social simulation model. This aspect is linked to the verification processes. Since we are referring to knowledge holders, our perspective is connected with the User community view.

Chapter 3

Planned Methodology

The methodology for this study is depicted in Figure 3.1. The analysis and steps for constructing the bullying model are presented in Section 3.1, while the steps for designing and developing the DSM tool are described in Section 3.2.

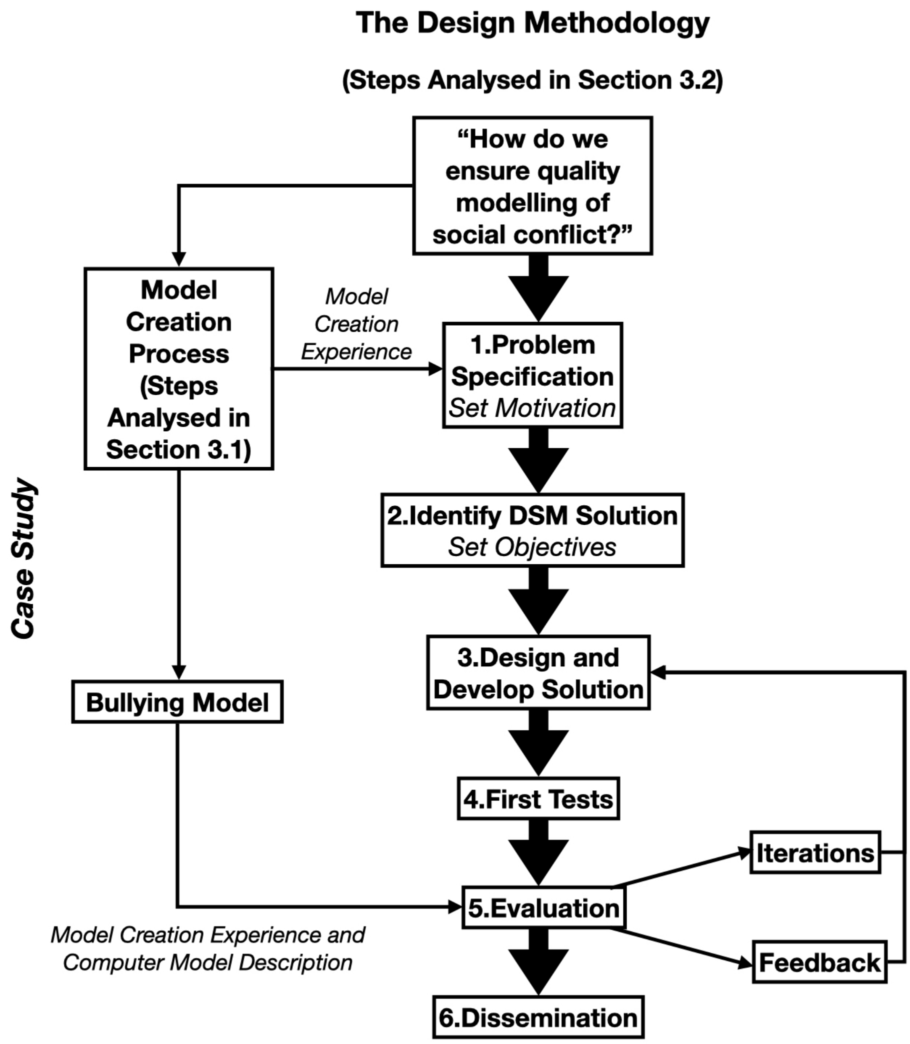


Figure 3.1: Steps in the Design Approach inspired by [2]

We decided to utilise the design methodology for this study because our ultimate objective was the DSM tool design and development. The design methodology is commonly used in engineering fields and has recently been applied in the field of Information Technology [2]. As Peffers et al. note in their paper [2], the design methodology differs from other traditional scientific methodologies, which tend to focus on understanding because it is geared towards creating solutions.

The design methodology, as conceptualised by Peffers et al. in their paper [2], consists of six steps and includes iterative processes shown in Figure 3.1 (Steps 1 to 6, Iterations and Feedback). As they mention in their paper, depending on the project goal, the starting point of a project may not necessarily be at step 1. For example, it is possible that an artifact already exists and can be repurposed for a different context, in which case the development phase is already underway or completed.

Our project aimed to address the potential misalignment between knowledge holder insights and the model implementation. Therefore, our first step was to analyse the problem using existing literature (Step 1 in Figure 3.1). Based on the state-of-the-art methods, identified research gaps, and outlined DSM principles, we would select an objective and solution for the design in step 2. The first version of the DSM tool would be outlined and developed in step 3. In step 4, the DSM tool would be used and tested for the first time. The tool would be evaluated with the case study model in step 5. The design methodology is generally an iterative process, and we expected to receive feedback at step 5 that would allow us to improve the design and create updates.

The case study typically enters the development phase during step 5. We planned to begin the case study earlier in the design process to evaluate the tool and gain a thorough understanding of the problem. The development of a social simulation model is a lengthy and complicated process, so we planned to tentatively identify a problem and solution before finishing the case study.

3.1 Building the bullying model

This section details the steps and processes we planned to follow to construct the bullying model. Section 3.1.1 examines the various collaboration options and outlines our preferred approach. Section 3.1.2 outlines the standard steps in building an ABM model and our specific plan for this project.

3.1.1 Collaborations

In a social simulation modelling project, collaborations often occur between the model team and knowledge holders and between the modelling team and stakeholders. The involvement of collaborators in the social simulation project is considered a “participatory approach” in the literature [55]. It is important to note that the roles of knowledge holders and stakeholders do not always overlap. Knowledge holders are individuals who have expertise in specific domains, either through direct experience or academic research. In some cases, such as in Action Research, researchers have both hands-on and academic experience. On the other hand, stakeholders are people responsible for or directly affected by interventions in the case study.

To better understand the distinction between these two roles, consider a project to prevent bullying in classrooms of School X. The teachers at School X would be regarded as knowledge holders and stakeholders since they have expertise in the subject matter and are directly affected by the potential interventions. On the other hand, the principal of School X would only be considered a stakeholder (assuming this person does not have teaching responsibilities). Researchers studying classroom bullying would be considered knowledge holders. If we were to conduct a modelling project on the topic of classroom bullying, we would seek insights from both teachers and researchers, while the principal would be responsible for making decisions based on the output of our model. It is important to note that this distinction is not commonly made in the literature. For example, Barreteau et al. refer to all collaborators as stakeholders in their work [55]. However, for the purposes of clarity, I will maintain the distinction throughout the text.

The modelling team should actively seek collaborations with knowledge holders to ensure the exchange of information and insights. It is safe to assume that it is not common for knowledge holders to initiate these collaborations, as most academic and non-academic individuals have busy schedules and may not be aware of relevant modelling projects. Furthermore, many modelling projects are not openly advertised until the first version is completed, making it difficult for potential knowledge holders to track them and participate. Therefore, the modelling team needs to take the lead in establishing collaborations with knowledge holders in order to benefit from their expertise and input.

Barreteau et al., in their article [55], have presented different collaboration styles based on the **activity** of collaborators and the **control** they can exert over the model usage. To be more specific, the activity directly affects the model content, while model usage refers to how the model affects reality and policy decisions. In other words, the control parameter is relevant for collaborators who are stakeholders. When in control, stakeholders can dismiss the model's validity and prevent its use. The first category, "Information-no control", describes a passive involvement of collaborators most likely to happen when there is a social simulation project for policy support where collaborators are stakeholders. The main distinction between this category and "complete lack of collaboration" is that there is at least some exchange of information and the possibility for passive cooperation to progress to a more interactive collaboration style eventually. The second category "Consultation-no control" refers to cases where knowledge holders provide insights but have no control over model usage.

The following two collaboration types "Dialogue-no control" and "Dialogue-control" imply a more active interaction between collaborators and modellers that could lead to more active participation in the model design. In a sense, collaborators can affect how their insights are used through feedback sessions. In the former case, collaborators are not given control over the use of the model; in the latter, they can affect how the outcomes will impact policy decisions. "Co building-no control", and "Co building-control" mean that collaborators are both involved in the model design and building process and can determine how their insights are implemented in the model to the best of their knowledge. Both parties are more likely to converge about the model and trust each other in these cases. The category "Co-building -control" gives collaborators the most power over the model creation and usage.

Other aspects that further diversify the collaboration possibilities are the patterns of interaction and the time of interaction [55]. To be more specific, there are two main patterns of interaction: "individual involvement" and "group involvement". Group involvement is further divided into "heterogeneous", or "homogeneous" groups. Then, interactions can occur at the start, middle, end, or at multiple stages during the project development.

When examining the matrix of different collaboration styles, certain combinations are more prevalent. For example, a presentation at a conference could be considered an "information-no control" case directed towards a heterogeneous group at the end of the project. In contrast, workshops used to create simulation models would involve "co-building" with or without control and occur multiple times throughout the project. Barreteau et al. have pointed out that the general use of the term "participatory approach" in different contexts can be misleading, as it can blur the distinction between passive attendance and active participation. Therefore, it is essential to distinguish between the different collaboration styles to understand the implications of active collaborations fully.

Finally, it is important to consider situations where there is no collaboration. In these cases, modellers rely on published theories or use the model to test their own theories. This approach is not problematic when the goal is to generate insights through analogy, as discussed in [44]. However, further consideration is needed when the goal is to establish a link between the model and reality.

We chose to adopt a participatory approach in developing our simulation model for three main reasons. First, we wanted to gain practical experience incorporating knowledge-holder insights into the model. Second, as our goal was to understand and support interventions, it was necessary to involve stakeholders. Finally, our team did not have any prior expertise in the topic of bullying, so we needed to rely on external input and assistance in building the model.

3.1.2 Model development

Building a social simulation model extends the typical social science research process as described in [30]. An overview of the model stages and processes in social simulation projects is given in Figure 3.2 (inspired by [31], [54] and [57]). In traditional social science research, the conceptual social model summarises the findings of a project [30]. However, within the computational social sciences and especially within the social simulation methodology, the conceptual model is defined in the early stages of the project.

Figure 3.2 illustrates how knowledge-holder insights inform the model creation process, either directly or indirectly. Within the context of participatory approaches, knowledge holders provide their insights directly to the modelling team, as outlined in Section 3.1.1. Additionally, knowledge holder insights can indirectly enter the model-creation process through surveys, interviews, theories in academic books, etc. It is also possible to build a conceptual model based on hypotheses made by the modelling team to test new theories. It is important to note that sometimes the modelling team does not include all the information it receives but instead selects some of it to limit the model's scope. Note that in this diagram, knowledge holders provide the modelling team with the connection to reality.

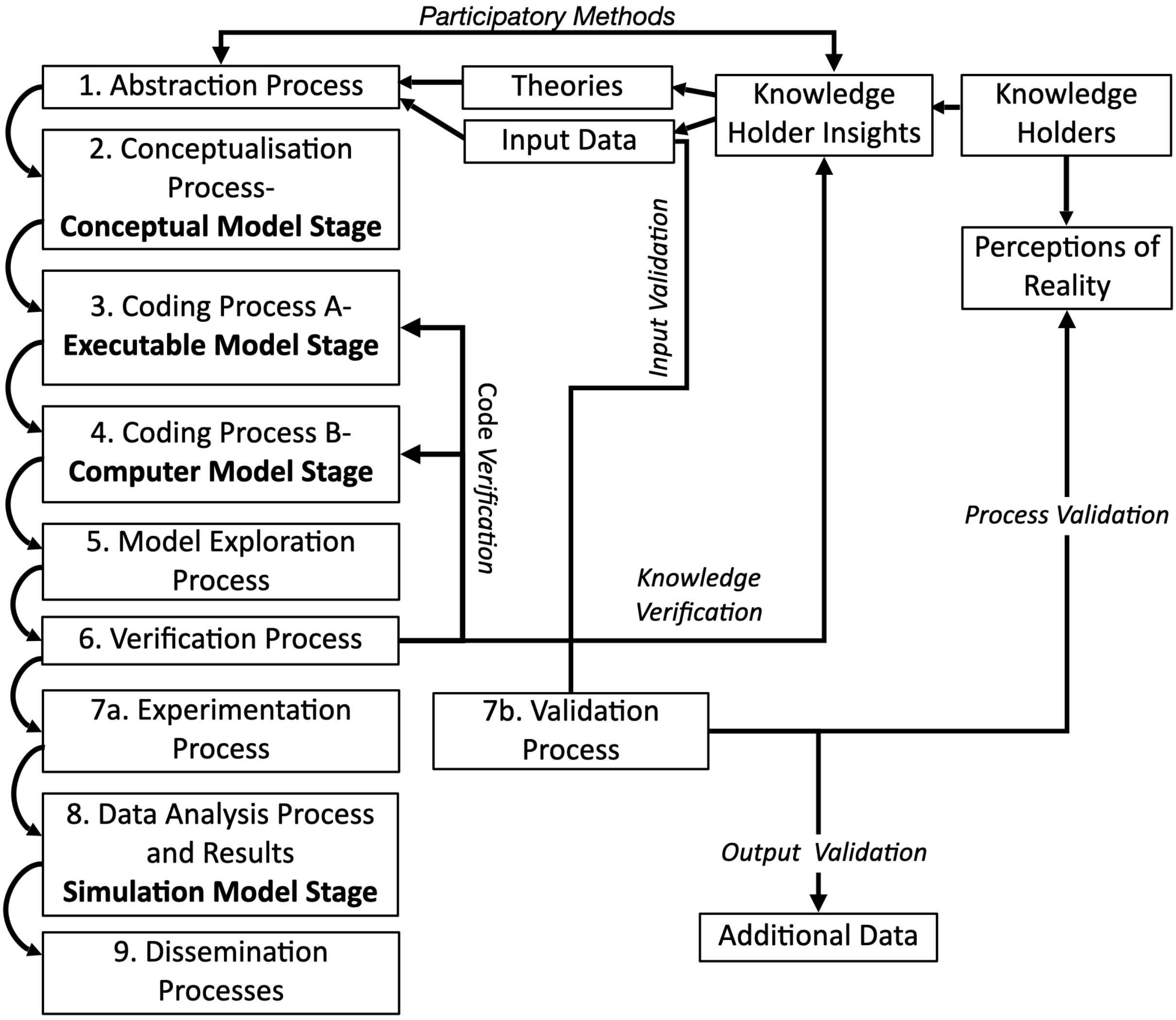


Figure 3.2: Model Stages and Flow of processes in a Social Simulation project

The social model is transformed in multiple stages in the social simulation methodology. The different stages of the model include the conceptual, executable, and the simulation social model (see Figure 3.2) [31]. The transitions are actualised with processes such as conceptualisation, coding, and model exploration and are further embellished by corrective processes such as the verification and validation processes [54, 57]. To connect this detailed analysis with the more generic analysis in Figure 1.1, Transition 1 (from the conceptual model to the computer model) includes processes 3 and 4 and Transition 2 (from the computer model to the simulation model) includes processes 5, 7 and 8. To connect with the problem statement, the operationalisation process includes processes 2 to 4. The conceptualisation processes include the selection of definitions, the model variables and parameters (stage 2 of Figure 3.2). During the conceptualisation phase, we must decide how to portray our input data, theories, and received knowledge insights to entities (general and environment), properties, behaviours, and model parameters (properties that do not belong to entities) according to the ABM framework. The conceptualisation process is followed by the coding process, marked as stage 3 in Figure 3.2.

The final part of the operationalisation process, consisting of the coding processes (Processes 3 and 4 of Figure 3.2), is critical. During this phase, we must specify how abstract concepts will be quantified. It is also a process where we can merge equation-based and conceptual models by selecting equations to portray the dynamic character of specific variables.

After completing the computer model, we must ensure that the code achieves what we envisioned in the conceptual phase. This is accomplished during what is called "the verification process" (Process 6 of Figure 3.2). It involves activities such as checking the code before the first simulations, performing unit tests and running simulations to get an impression of the artificial society that emerges (denoted as Model Exploration in Figure 3.2) [54]. Then, the code is updated, and coding bugs are fixed if needed. According to [54], the second part of the verification process consists of activities to ensure that the model is consistent with the data and theories that inspired the conceptual model. In other words, the second part of the verification process includes checks between the different knowledge insights used to create the conceptual and the computer models. For better clarity, I will refer to the first part of the verification process as "Code Verification" and the second part as "Knowledge Verification". It is important to consider researchers include what I defined as "Knowledge Verification" in the validation processes [55, 58–60].

Following the verification process are the experimentation and validation processes (7A and 7B of Figure 3.2). I distinguish between input and output data and refer to dependent and independent variables to better understand these processes. The model properties and parameters make up the model variables. Most of the model variables are defined in the conceptualisation phase. Some model variables can also be defined to fill conceptual gaps in order to create the computer model [31]. Most variables change with the progression of the simulations based on the coded model behaviours. The input simulation data, which are part of the input data of the model creation process (see Figure 3.2), set the values for the independent variables at the beginning of the simulations. If we cannot match independent variables with our input data, we either calibrate the missing values [61] or initialise the independent variables using different scenarios. To understand our model behaviour, we collect some or all the variable values for each time step or for targeted time steps. The collected variables make up the output data. Overall, each simulation is identified by the initial model variable values, the simulation time, the output data and the random seed. The random seed is the collection of random values used throughout the simulation. As an example, suppose we use a model parameter that changes its value based on a particular function. If the function updates its state based on randomness, then we use one random value from the seed every time the function is called. The same random seed always leads to the same output data.

The goal of validation is to ensure that our simulation acceptably matches reality. In this phase of the model creation process, it is also possible to calibrate model parameters that cannot be matched with collected data. Graebner in [54] identifies four validation types:

- input data validation, where we examine if we correctly designed the data that set the initial conditions of the simulations

- process validation, where we assess whether the outcoming model mechanisms are "reasonably" compared to "reality"

- descriptive output validation to ensure the model output matches data collected from the targeted system (but preferably not the ones we used to create the model)
- predictive output validation to evaluate if the model output successfully predicted (when relevant) aspects of reality

Each validation process is achieved with activities such as comparing simulation outcomes with data and reviewing simulation outcomes by domain experts [57]. This often happens in parallel with the experimentation phase.

Experimentation means we run many simulations starting with different initial model variable values to explore the model (see stage 5 in Figure 3.2). Sometimes, we run simulations with the same initial model variable values multiple times to account for the effect of different random seeds and to understand the system's complexity. Gaining insights from experimentation is not always straightforward. Sometimes, we must analyse the different outcome simulation data to understand how they translate to the observed reality. Most of the time, it is impossible to explore our simulation models fully, and we restrict ourselves to specific scenarios based on reasoning. The conclusions and knowledge from the data analysis give us an understanding of the simulation model. After we have collected insights from our simulation outcomes, we communicate the insights to stakeholders.

3.1.3 Model Development Steps

For this project, we decided to work with a mix of collaboration types based on the phase of the project. Before reaching out to potential collaborators, we had to consider which general theme would attract the most local and external collaborators (Step 1). After this decision, we would reach out to both knowledge holders and stakeholders (Step 2). By engaging with stakeholders, we aimed to understand their needs and use that information to guide the focus of our modelling efforts (Step 3). Eventually, the goal was for our stakeholders to use the model output for interventions. Our main plan was to give stakeholders complete control over the use of our model. Regarding local knowledge holders, we aimed to arrange consultation meetings about various aspects of the model (Step 4). Depending on their enthusiasm level, we would consider more interactive collaborations, such as model co-creation. Finally, we planned to seek out external knowledge holders through conferences and seminars to complement our modelling needs and secure diverse knowledge insights.

The participatory approach, in combination with the model creation process analysed in Section 3.1.2 and depicted in Figure 3.2, resulted in the following steps:

- **Step 1** Specification of the Case Study. Bullying takes place in different contexts. Our first step was to select an appealing case study context to pitch to the local stakeholders.
- **Step 2** Establishment of Collaborations. In this step, the plan was to reach out to potential collaborators within our network and establish a participatory approach to the model creation process.
- **Step 3** Feedback Meeting - Selection of Topic. During our initial feedback meetings, we planned to specify the modelling topic to serve stakeholders' specific interests.

- **Step 4** Feedback Meeting - Conceptualisation processes. In the next round of feedback meetings, the planned goal was to form a conceptual model description for bullying with the help of our collaborators (Step 2 of Figure 3.2).

- **Step 5** Conceptual Model Design. The plan for this stage was to conclude the conceptualisation processes, with the potential supplementation of literature reviews and data collection processes.

- **Step 6** Coding. The next planned step was the coding processes, which corresponds to steps 3 and 4 of Figure 3.2. Since our team had experienced programmers, we would not need external support to finalise the computer model.

- **Step 7** Preliminary model version. Coding processes are followed by verification checks to ensure the code does not contain unintentional errors (Step 6 of Figure 3.2). In combination with the verification checks, we planned a preliminary model exploration to get a grasp of our model behaviour (Step 5 of Figure 3.2).

- **Step 8** Feedback for Knowledge Verification round and Process Validation. This verification round includes Knowledge Verification processes mentioned in Figure 3.2. Errors found during Code Verification lead to changes in the code, while errors during Knowledge Verification lead to updates in the conceptual model and new coding activities. At this stage of the model process, we planned to seek advice from external knowledge holders during formal meetings, such as conferences. We expected to communicate with people whose theories and knowledge insights were included in our conceptual model. We would request these knowledge holders to perform Knowledge Verification on our conceptual model and Process Validation on the model's behaviour (also mentioned as face validation). Assuming a fruitful data collection process, we planned Input and Output validation processes (See Figure 3.2). When the model is lacking and new additions to correct the model are identified, the conceptual model is updated, and the process starts again from step 5.

- **Step 9** Agreement for the Final Model Version. To finalise the simulation model, knowledge holders and the modelling team had to approve the conceptual model description and the outcome in model behaviour. If otherwise indicated, we would perform additional conceptualisation processes and work from that stage till step 9 again.

- **Step 10** Implications for Interventions. During this step, we would analyse the results of our verified and validated simulation model and would draw conclusions on interventions (Step 8 of Figure 3.2). Expected results included explaining the mixed intervention results and potential suggestions for successful interventions. The results would be disseminated in relevant outlets such as the Antibullying Forum (Step 9 of Figure 3.2).

- **Step 11** Feedback for the DSM tool. The experience with the case would help us analyse the problem statement to gain a fuller perspective on the potential DSM solution.

The model creation process is supplemented with the following methods:

- Literature search. This method is used as a supplement to knowledge holder insights.
- Data Collection. We planned both private and informal data collection methods. Private data collection methods mean we collect private data. For this type of data, we need permission from the Norwegian Data Protection Agency (NSD) and must handle the data securely, anonymise them and eventually erase them. Regarding the data collection methods, both quantitative (surveys) and qualitative methods (observations and interviews) were planned in the project.
- Reflection Processes. The Action Research Methodology inspired the inclusion of reflection phases in our methods [62].

3.2 Domain Specific Modelling

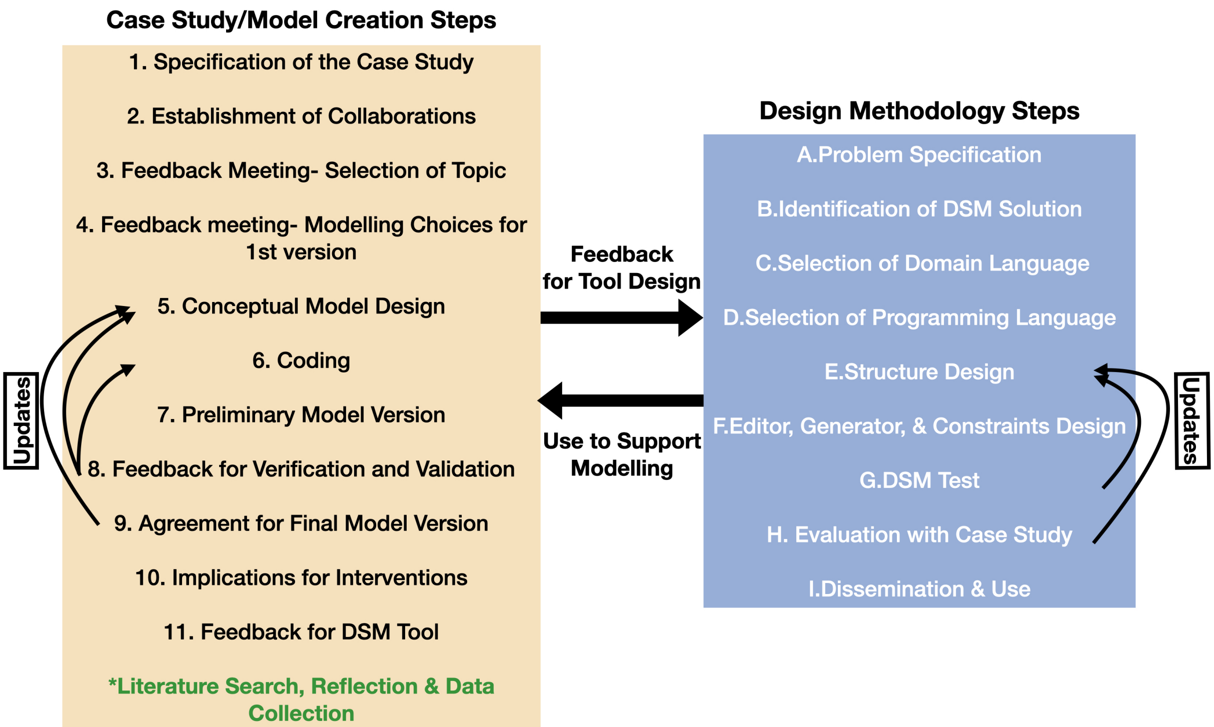


Figure 3.3: Planned Methodologies with Detailed Steps

Figure 3.3 collects the Model development steps (Steps 1-11 of Section 3.1.2) and connects them with the design steps using the template and connection points of Figure 3.1. We decided to follow the design methodology steps from Peffers et al. [2].

- **Step A** Problem Specification. The first step of the design process is problem specification. In our case, we started with a problem statement but evaluated that the problem was not sufficiently analysed to fully understand the requirements for a DSM tool. As a result, we planned to fully explore the problem using the case study experience, additional literature and reflection processes.

- **Step B** Identification of a DSM Solution. The solution had to address the problem adequately. We planned to identify a tentative solution based on the initial problem statement at the start of the project.

- **Step C and D** Selection of a Domain Language. Each DSM tool consists of a Domain Language, a Programming Language, and the definition of transformations from one to the other (see Figure 1.2). The nature of the identified problem would inform the selection of the domain and programming language.

- **Step E** Structure Design. The first step of the language design is the structure design, i.e. the concepts of the language [63].

- **Step F** Editor, Generator, and Constraints Design. After defining the concepts, we planned to determine the remaining language aspects: the editor, generator, and constraints [63].

- **Step G** DSM tests. After completing a sufficient amount of the language implementation, we would conduct the first language tests. In these tests, it is essential to choose easy models [1] for implementation and gradually move to more challenging ones. Depending on the DSM tool performance, we planned to update the domain language design and implementation according to the principles of Agile Programming [64]. In other words, we would extend the initial language design to express the test models sufficiently. Once this was achieved, we would move to more complex examples and stop at the point we evaluated the language structure adequately developed to accommodate most models.

- **Step H** Evaluation with Case Study. In step H, we would test the DSM tool with the case study simulation model to receive feedback on the tool's state and get support for the case study.

- **Step I** Dissemination and Use. The last step of the design methodology is the dissemination of the work we have done. The primary target group for the DSM tool would be researchers interested in using/already using the Social Simulation methodology.

[1] The easy part relates to the small number of parameters used in the model

Chapter 4

Exploring the Bullying case study

As mentioned in 1.2.1, when we started the project, there was a lot of momentum regarding bullying at our University. Since both the initiatives by the rector and Ingrid Lund's survey targeted University bullying, we further specified our case study from bullying to Bullying at the University of Agder to increase our collaboration chances (Step 1 of Figure 3.3). In Section 4.1, you will read general information about the case study to understand our model context better. In the next section 4.2, you will read about the collaborations we established. The rest of the Sections document our model development process and the interesting results concerning the concept of bullying.

4.1 Specification of the case study (Step 1)

Before I describe the University environment on bullying issues, it is important to acknowledge the Norwegian environment. Norway has two laws that force Universities to take incidents of bullying seriously. First, legislation requires Universities to be safe learning environments [19]. Thus, bullying can be considered illegal as it disrupts feelings of safety. Universities are not only educational institutes but also big employers. As a result, it is subject to the Work Environment Act [65]. One part of the Work Environment Act requests anyone who experiences or witnesses events of "wrongdoing", such as bullying, to report them. Companies and institutions need to accommodate a reporting system to enable this action.

The "Speak Up" report system of UiA is one of the most prominent University initiatives to deal with bullying issues. A very telling sign of the importance of the report system is its placement at the navigation bar of UiA's website (see Figure 4.1). In other words, everyone with access to the website can immediately see and use the report system. Answering the question of who is the target user, it would be anyone who has experienced or witnessed "wrongdoing" by someone affiliated with UiA [66,67]. The person who reports can use different formats and needs to support claims with documentation [68,69]. Anonymous reports are also possible [67], but there is also protection for the people who report using their identity [68,70]. Of course, depending on how the case progresses, the person who reports might also need to defend the claims [68].

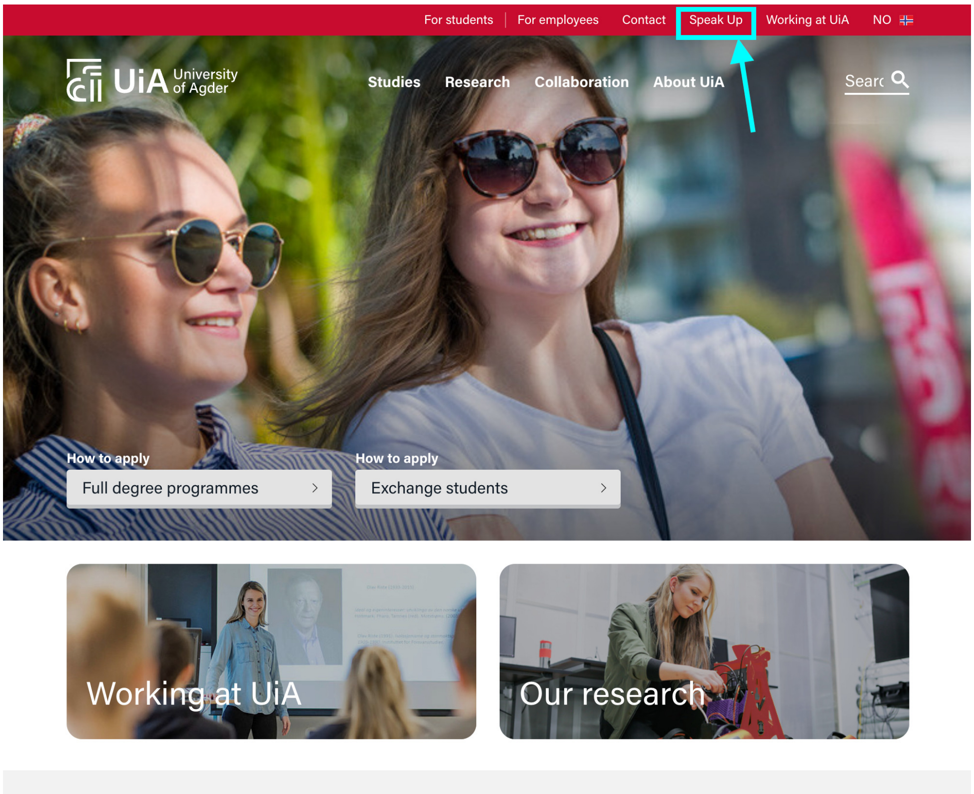


Figure 4.1: The frontpage of UiA website

The Speak Up system allows people who report to choose who is to take on the case [71]. Examples of case handlers include student representatives, faculty leaders, and the human resources department. Strategies for handling the claims are not detailed on the website. However, it is mentioned that after the report is finalised, a receipt is given [68]. Then possible scenarios include a written warning. If the warning does not improve the situation, options include the dismissal of the accused party, which can be very serious given that this is later communicated to other Universities [72].

Apart from the Speak Up system, there are also preventative measures. Having acted as an instructor at the University of Agder, my experiences have shown me that instructors try to ease classroom tension, act mindfully around group work, and explicitly request students to respect one another. A more tangible preventative measure is the FYSE (First Year Study Environment) program. When our project started, the FYSE group was in its early phases. The idea behind the FYSE project was to assist first-year bachelor students with aspects of student life and equip them for their years of study. To achieve this goal, each 1st-year bachelor student had the opportunity to join a FYSE group. Each FYSE group was composed of about ten students and led by a mentor. The group would function as a study team and an opportunity to socialise. Mentors would organise meetings to talk with students about their concerns, to provide information or even skills, and to have fun. If a conflict arose in the FYSE team or between a FYSE group member and someone external to the group, FYSE mentors had the opportunity to provide advice or to take action to resolve the conflict.

Overall, the Norwegian and University environment discouraged bullying, and many measures were placed to facilitate conflict resolution.

4.2 Establishing collaborations (Step 2)

A failure to establish collaborations could mean we had to discard bullying as a modelling theme and choose another type of social conflict. In addition, we needed the collaborator's input to crystallise our modelling subject and to create a first model version that could be further updated. Finally, we approached the bullying theme with interest in interventions, which meant finding an actual arena to design and implement interventions was essential. As a result, reaching out to potential collaborators was critical.

We connected with different stakeholders and knowledge holders through our university network. Our primary knowledge holder was Professor Ingrid Lund, who had conducted the bullying survey. Ingrid was open from the beginning and keen to share her knowledge and contribute to the project. In the course of the CTMSC project, I met more UiA researchers with interest in bullying. However, most of them joined the University team later than me. As a result, we met and discussed bullying after the project was already well-defined. The rest of the collaborators were mainly stakeholders; some had first-hand observations and experiences as case handlers. On the higher organisational levels, there was the university team working on behalf of the rector. In addition, we interacted with some of the Human Resources personnel. On a lower organisational level, we engaged with program coordinators. Finally, we came in contact with the FYSE program (First Year Study Environment). Overall, we decided to align with the FYSE program regarding project goals while retaining connection with program coordinators to allow the implementation of interventions on a program level.

4.3 Consultations and Stakeholder Meetings (Step 3)

4.3.1 Knowledge Holder Consultations

Our first meetings with our primary knowledge holder, Ingrid Lund, aimed at providing a good orientation of bullying research. One of the challenges in selecting bullying as a theme was our group's lack of expertise in bullying research. In addition, our project timeline and the other project requirements meant that we could not invest much time researching the massive literature on the topic. As a result, we depended on expert consultations to identify theories for our model and later to update it. From these early consultations, we received two important insights about bullying. First, a new way of thinking about bullying has emerged recently. Second, at the time of our meetings, there was no explicit theory of bullying.

In 2014, Schott and Søndergaard proposed a new definition of bullying [73]. Before that, the most prominent definition of bullying by Dan Olweus stated that: "A person is being bullied or victimised when he or she is exposed, repeatedly and over time, to negative actions on the part of one or more other persons" [74]. Subsequent research has highlighted important aspects not captured by the Olweus definition. For instance, group procedures also take place in these events. Literature shows that in addition to the "bully" and the "victim", we also observe roles such as supporters of the bully, outsiders or bystanders, and defenders of the victim [75]. These roles are correlated with the social status of the participants. In addition, the above definition raises the question of what constitutes a negative action. In many cases, the negative is in the eye of the beholder. The Schott and Søndergaard definition offers a broader conceptualisation of bullying, describing it as a process of marginalisation in which the persons involved do not necessarily perceive or intend to create a negative aspect on their interactions [73]. Consequently, we decided to proceed with the new definition. More details about the Schott and Søndergaard definition selection appear in Paper B.

Apart from the bullying perspective, the insights by our knowledge holder implied that we needed to give our model an explanatory purpose. A first literature search on bullying showed that most bullying studies have a correlational and not explanatory character, and theories of bullying were not prominent. This was later confirmed at the keynote lecture at the first AntiBullying Forum we attended. In addition to the lack of theories, research on University Bullying was limited at the start of our project. Ingrid Lund herself had researched University bullying to conduct the survey that alerted us to the situation of bullying at Norwegian Universities, but her main domain was bullying in younger people. For our project, we needed to test different theories that could help us explain the situations we were to face.

4.3.2 Stakeholder Meetings

From our stakeholder meetings, we expected to specify our modelling subject and secure an arena of interventions with opportunities for data collection. Given the rich structure of the FYSE project, the eagerness of project leaders, and the opportunity for data collection, we decided to invest most of our energy in a collaboration with FYSE stakeholders. Part of the FYSE project's agenda was to ensure students' successful graduation. In other words, they wanted to reduce dropout rates. Their interest in bullying was to the extent that it was potentially driving some students outside their studies. From our perspective, being on the receiving end of bullying behaviour seemed like a plausible hypothesis for eventually dropping out. In fact, there has been research linking bullying with school drop-outs [16]. So, understanding and helping foster positive student interactions aligned with our interests. At the same time, during some FYSE meetings, mentors would give students short surveys to evaluate their work. These informal surveys offered the best opportunity for data collection.

The result of our meetings with the FYSE group signalled our interest in researching bullying from the perspective of group work and interactions. Anecdotal evidence and my interactions with bachelor and master students further fortified the need for a focus on group work, as many complaints arose in instances where students had to work together. In addition, we contacted group mentors, some of whom were also open to provide insights from their observations on student interactions.

At this point, I should also mention that we initiated a round of interventions, informal data collection, and a design for private data collection. However, for reasons that will be analysed in Section 4.6, I chose not to enclose them in the dissertation as they did not contribute to our overall results and conclusions.

4.4 Modelling Choices and First Version (Steps 4-7)

The purpose of the first model version was to provide a ground for further feedback from knowledge holders. Even though we had engaged with many stakeholders, we needed more support from bullying experts to develop a theory of bullying. So the general plan was to present a tangible model with preliminary results to an audience of bullying experts, communicate the model function and results, and get feedback. We chose the Antibullying Forum 2019 as our venue. The Antibullying Forum is an interdisciplinary conference held every two years to connect scholars from different backgrounds and a shared passion for preventing bullying.

The version we presented at the Antibullying Forum is very similar to the one we published in Paper B [76]. To provide a good indication of the language used to communicate our model contents, I have also attached the poster for the Antibullying forum in the Appendix Section. It appears as Poster F. The model code and documentation using the ODD protocol can be found on GitHub (https://github.com/themisdx/MARG). Some key points from our model creation process (Paper B) are the following:

- **Operationalisation of bullying (Part 1)** The new definition of bullying, which states that bullying is: "the intensification of the processes of marginalisation that occur in the context of dynamics of inclusion-exclusion, which shape groups. Bullying happens when physical, social, or symbolic exclusion becomes extreme, regardless of whether such exclusion is experienced or intended" [73] required further analysis to become operationalised. We decided to focus on the inclusion-exclusion dynamics aspect of the definition.

- **Evaluation of Previous Work** Existing Agent-Based Models on bullying operate using the Olweus definition. Thus, they could not act as a starting point for our model creation. In addition to searching the literature for bullying models, we tracked models related to inclusion and exclusion fitting our context. We found one model related to exclusion in groups [77]. However, we decided to discard it. One model aspect, the homogenisation processes, meant that our model would lose its explanatory power. Overall, the lack of suitable existing models meant we had to build the model from scratch.

- **Modelling Choices for Case Study** Limited by our publication space, the context of the model is not thoroughly analysed in Paper B. Some of our modelling choices were linked with our case study. Firstly, inspired by the FYSE project, we proceeded to specify inclusion and exclusion in the context of student groups in the educational environment. Based on our discussions with the different stakeholders, we decided on a set with two types of interactions: forced and hang out. The forced interactions simulated supervised spaces, such as the classroom and group rooms with mentors, where students could not avoid interacting with other students. This is not true on other occasions, such as during breaks. Our interest in unsupervised interactions (the hang-out interactions) was reinforced by discussions we had with stakeholders. They had mentioned that students often exercised their freedom to reject interactions with other students during those "hang out interactions". This insight was not evaluated with formal data collection or data analysis. However, it could still work as an indication and hypothesis that could be later assessed with our model. During the hang-out interactions, students can opt out and refuse communication with the other student. When there is an interaction, either forced or free, we identified two experiences: positive and negative. For this model version, we decided to focus on dyadic interaction. At each time step, we allowed each student to initiate one free interaction and to engage in a forced interaction.

- **Operationalisation of bullying (Part 2)** We further operationalised bullying by expressing exclusion as the ratio of negative and refused interactions to the total number of interactions. When the ratio exceeded a number, which we assigned to be 80%, we called the exclusion intensified exclusion and labelled it as "Marginalisation", which led to naming our model "MARG".

- **Selection of Theories and More Choices** One of the biggest challenges was the selection of theories for our model. The literature search of inclusion and exclusion processes in groups led us to a book entitled "The Social Psychology of Groups" by Thibault and Kelly [78]. The presented theory suited our interpretation of the selected bullying definition well. The theory suggests we gain from interactions when we have compatible partners. In contrast, we become depleted when we are not compatible. We interpreted the former as "positive interactions", and the latter as "negative interactions".

  Upon reading theories about the influence of norms on bullying behaviour, we decided that students would evaluate compatibility based on a personalised set of values influenced by culture. Each student would evaluate the interaction positively or negatively based on the compatibility evaluation. We assigned each student several traits, some visible at first sight and some exposed during interactions. Compatibility evaluations fluctuated over time depending on which characteristics each student acknowledged about the interaction partner. As a result, it was possible for a student to have negative interactions with another student initially but to proceed to have positive interactions later in the model because new characteristics coming to the surface were compatible with personal values. Finally, we also introduced a variable of attraction that fluctuates based on whether previous interactions have been positive or negative. The consequence was that unattractive interaction partners were more likely to be refused. Refusal during hang-out interaction also had a randomness component.

  One of the most interesting aspects of our model was the perceptual character of the interaction. Each student evaluated an interaction based on their own set of values. Consequently, an interaction, be it free or hang out, could be positive for one student and negative for another. This way, we integrated parts of the bullying definition referring to the importance of perception and intention in bullying incidents.

- **Repurposing of our First Version** Our interpretation of theories was unverified and our model could not be validated before our Antibullying Forum presentation. Thus, we used this first model version as a tool to think about bullying. This model type has been categorised as analogies [44].

- **Coding and Data Analysis** We coded the model in NetLogo and ran different unit tests, agent tests [57], and extreme scenarios as part of the Code Verification process. Next, we conducted Model Exploration (see Figure 3.2) with more scenarios accounting for different cultures. We did not conduct calibration as the first model version was vague. We performed data analysis on the simulation results.

- **Conclusions on Ethics of Bullying** For a detailed presentation and discussion of our model results, please refer to our paper, Paper B. From my perspective, one of the most exciting reflections from the model results was the ethical implications of norms. Essentially, we were modelling students with values aligned with cultural values, and we ended up observing the marginalisation of some of the students. In our model, marginalisation was not expressed with violent behaviours. In fact, we did not specify how negative and refused interactions were realised. However, since the new perspective on bullying focused on the perception of marginalisation, it did not matter. That an acceptable value can lead to a behaviour we do not tolerate sounds very counterintuitive. It suggests that before we attempt to fix behaviours, we should consider our values. Another fun fact coming from the model was the randomness in perceiving bullying. In other words, students could perceive negative interactions in cases where their interaction partners perceived positive ones. This could further escalate feelings of marginalisation. As a result, in addition to our ethical considerations, we should also consider how much of our perception is constructed. Finally, the last part raises many questions about who should be regarded as a bully. Should I be considered your bully if I perceive a positive interaction with you but you perceive a negative one? If I am refusing to interact with you because we are incompatible, am I marginalising you? These types of questions initiated great discussions with our stakeholders. One example that different people have noticed is the following: a misbehaving student who is later disliked and excluded. Our model captures these types of ethical problems. On the one hand, the misbehaving student could be considered a bully. On the other part, the group that decides to exclude the student are acting as bullies too.

4.5 Feedback from the Antibullying forum (Step 8)

The Antibullying Forum was a very vibrant conference. People with diverse backgrounds united to contribute to solutions for bullying issues. During the keynote session, it was confirmed that bullying did not have a theory. In addition, attendance in different sections connected me with discussions about the effect of norms on the expression of bullying behaviours.

We had two main expectations from our involvement: verifying and validating the existing model components and identifying missing and unintuitive parts. The theory we used in the model was first published in 1959. Thus we did not expect to verify its implementation with the people who created it. Also, at the time, we did not know that it was later extended and renamed. However, we could still attempt to understand how intuitive our model components were (knowledge verification), whether the results made sense (process validation) and what we had to include in the model to secure a good representation of bullying dynamics.

I followed two strategies to get feedback from bullying experts: asking questions in sessions and actively requesting feedback upon explaining our model. These resulted in a collection of insights from bullying experts from different fields. Table 4.1 [1] shows the different suggestions for updates from the various bullying experts at the Antibullying Forum. Note that most of the recommendations are not in the received format. I have edited and grouped them into five categories for better presentation. More information about the feedback received is presented in Paper C.

Table 4.1: Input grouped in categories retrieved from Paper C and reproduced with permission from Springer Nature

Personality	Psychological Needs	Personal skills	Contex	Social
Personal tendency to isolate Resistance to belonging Aggressiveness	General acceptance Acceptance from friends Belonging	Self-efficacy Sociometric status and perceived popularity Numb Blindness Connection and disconnection from self and others Questioning personal perception Social and Emotional Learning Filtering information Goal setting	Friendship networks Teacher as agents	Social capital Social impact Social influence Peer influence Social Norms Effect of role models Effect of leaders

4.6 Reflection results and Informal Interviews (Step 9)

A period of indecision followed the excitement of the Antibullying Forum. What initiated the indecision, was the number of suggested updates. Each one of them meant one or more extra properties or behaviours for MARG. However, there was not a clear direction for what was the most meaningful one. Aside from the indecision, probably ignited by anecdotal evidence, a sort of research intuition to take a step back and reflect kicked in. At that point, it was not possible to follow both my research instinct and at the same time continue with the implementation of the suggestions because of the project timeline. Because of my personal inclination, I decided

[1] originally published in: Themis Dimitra Xanthopoulou, Andreas Prinz, and F. LeRon Shults. The problem with bullying: Lessons learned from modelling marginalization with diverse stakeholders. In Marcin Czupryna and Bogumil Kamiński, editors, Advances in Social Simulation, Springer Proceedings in Complexity, pages 289–300. 2022, Springer Nature

Figure 4.2: The Sphynx cat and the Oncilla

to follow through with a reflection phase to track and analyse the source of confusion. The reflection phase led to a reconsideration of the bullying concept. The results of the reflection phase have been analysed in Paper C [79]. In the next paragraphs, you will read a short summary of what is presented in Paper C embellished with examples.

- **Concepts and Words** We distinguish between concepts, which are mental representations, and words, which are symbols of concepts. From this perspective, "bullying" is a word used as a symbol of behavioural concepts.

- **Concept Fuzziness** We define "concept fuzziness" as the existence of grey areas in categorising instances of concepts. To understand this, think of the example of cats. If you see for the first time a sphinx cat (see the left side of Figure 4.2), and you are unaware of this cat breed, you might not realise that it is actually a cat. On the other hand, if you see an Oncilla (see the right side [2] of Figure 4.2), you may identify it as a cat with spots.

- **Word Ambiguity** Word ambiguity happens when we use the same word to point to one or more concepts. To understand this, think of the word "crane". Crane is used to denote both a bird species and a construction machine (see Figure 4.3).

- **Bullying is Ambiguous** The reflection process, in combination with additional literature research and the conduction of informal interviews, showed us that bullying is ambiguous. However, bullying is ambiguous among people and not for the same person. In other

[2]The image was retrieved from "https://en.wikipedia.org/wiki/Oncilla" and is copyrighted under CC BY-SA 3.0 (see https://creativecommons.org/licenses/by-sa/3.0/ for copyright information)

Figure 4.3: Two very different types of cranes

words, for you or me, bullying may be just fuzzy. However, what the word means to me might be very different from what it means to you. This gives the term a "group ambiguity" property.

Even though some bullying experts expressed difficulty dealing with the concept of bullying, its ambiguity is not acknowledged. One reason lies in this "group ambiguity" property. It is much easier to identify the ambiguity of the word crane, for which each individual acknowledges the dual meaning. Another reason for the difficulty in realising the ambiguity of bullying is that it almost always points to concepts of behaviours during interactions. As a result, in contrast with the word "crane", which points to two concepts without associations, the ambiguity of bullying remains disguised.

- **Evidence of Ambiguity-Literature Search** Academic knowledge holders have been researching bullying with the use of the definitions. As mentioned earlier in Section 4.3.1, a new definition has emerged in recent years. The second definition introduces a level of disagreement on what constitutes bullying. When bullying researchers collect data, they typically use surveys, interviews and sometimes naturalistic data. As recorded in Paper C [79], surveys do not clarify the definitions sufficiently, the method for measuring bullying in interviews is not precise, and observations also rely on their own interpretations of aggression and power imbalance. At the same time, different studies (see [80, 81]) have shown that non-academic knowledge holders' understanding of bullying deviates from the definitions. Finally, a very interesting read on the history of the bullying concept shows the expansion of the concept over the years, with the author stating at one point: "The concept becomes everything and therefore means nothing" [82].

- **Evidence of Ambiguity-Informal Interviews** During our Informal interviews, we inquired on perceptions of bullying. Our sample was UiA employees from different organisational levels. We presented the main behaviour modelled in MARG and asked the interviewees if they considered it as bullying. We received mixed answers. Consequently, even within one institute, there is no shared understanding of what behaviours constitute bullying.

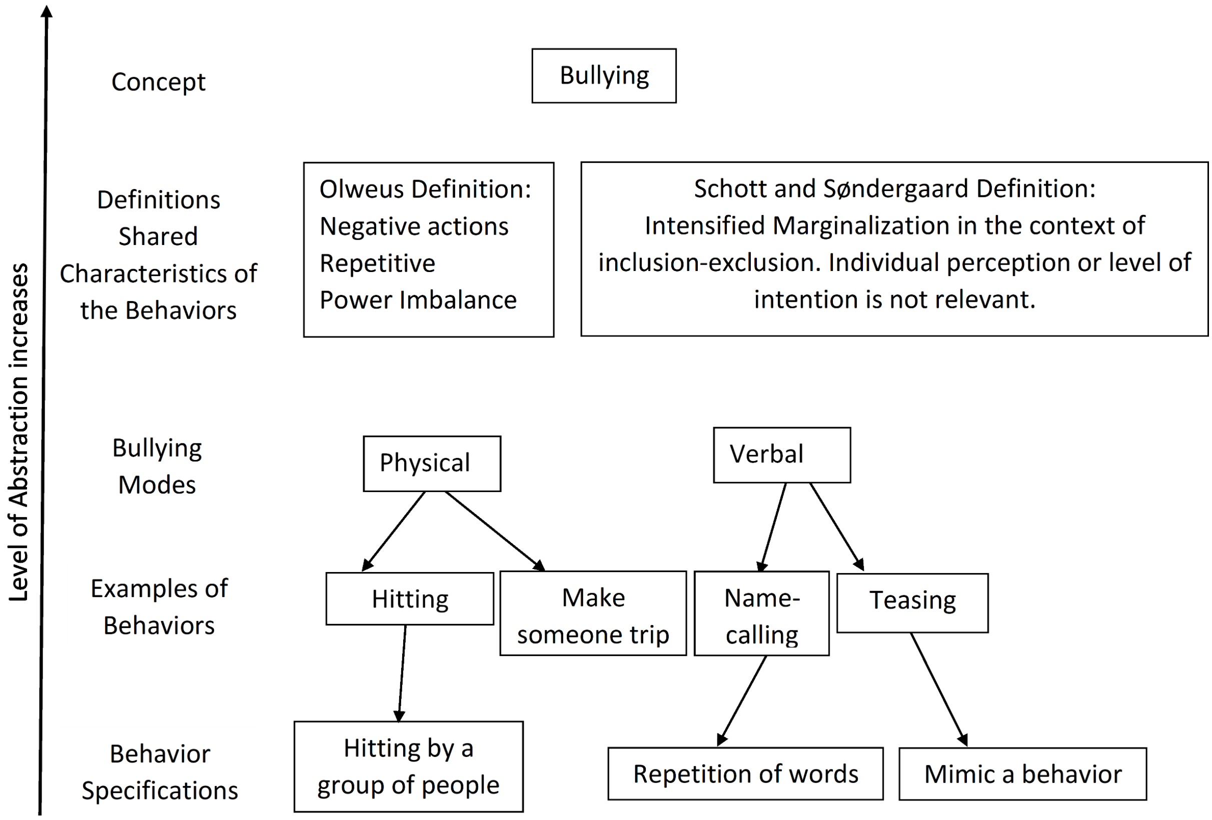


Figure 4.4: Bullying at Different Levels of Abstraction (Retrieved from Paper C and reproduced with permission from Springer Nature)

- **Analysis** We analysed bullying using different abstraction levels (see Figure 4.4 [3]). The highest level is the concept itself. On a lower abstraction, we find definitions of bullying. On a similar level, there are popular modes of bullying, such as physical and verbal. When we reach the level of abstraction of behaviours, descriptions become clearer. For instance, we do not expect hitting or name-calling to create the same confusion as bullying. However, behaviours can become more concrete with more details, such as "hitting by a group of people".

- **Reconsidering the Concept of Bullying** Bullying researchers tend to operate using higher levels of abstractions, while non-academic stakeholders observe behaviours. One hypothesis we made in Paper C is that bullying has become a concept of ethical judgement rather than a concept of behaviours. This can explain some subjective criteria for categorising behaviours as bullying observed in the literature (see [79]). Moreover, everyone agrees that bullying behaviours are unethical. So my current hypothesis is that non-academics label some of their observations as bullying based on how things look from an ethical standpoint. The result is the apparent disagreements between academics and non-academics on what constitutes bullying. When we model, we rely

[3]originally published in: Themis Dimitra Xanthopoulou, Andreas Prinz, and F. LeRon Shults. The problem with bullying: Lessons learned from modelling marginalization with diverse stakeholders. In Marcin Czupryna and Bogumil Kamiński, editors, Advances in Social Simulation, Springer Proceedings in Complexity, pages 289–300. 2022, Springer Nature

> on academic sources and data from non-academic knowledge holders. We eventually want stakeholders to use our model conclusions in explanatory and predictive models. However, we risk misunderstandings when we communicate using the concept instead of descriptions of concrete behaviours. This is not helpful because model documentation is not easily readable by everyone. Such a risk should not be dismissed, as model results could influence interventions. At the end of Paper C, we suggested using concrete behaviours instead of abstract concepts when communicating with knowledge holders and stakeholders. Finally, we highlighted the need to establish checks for modelling concepts before the start of the model creation process.

Our conclusions meant that our modelling process would benefit from removing the concept of bullying from our communications with the knowledge holders. For example, at the Antibullying Forum, instead of presenting our model as a bullying model with a focus on marginalisation, we would present it as a model of "X behaviour" in the context of group work of university students. Before using the name of the behaviour, we would have run tests to ensure that it is not an ambiguous term. Overall, behaviour X would be on the lowest level of abstraction of Figure 4.4. This type of detail would have helped experts in the field to provide feedback on our modelling topic without inserting their interpretation of bullying. We would probably gain less traction because it would not be as popular as bullying and with less literature attached. After all, fewer people would have input for it. However, the feedback would be more relevant to understand the phenomenon and proceed to help our collaborators. Finally, we would avoid using the word bullying with our stakeholders to clarify the specific cases in which our modelling insights should be applied.

In Computer Science, the issues of term ambiguity and concept fuzziness have been thoroughly discussed. Unger and Ciminano have identified various types of ambiguities in their research [83], including structural ambiguities which lead to different parsing and alternate meanings, instances where the same ontological entity is represented under different names and situations where the same name implies different meanings for different sources. Other ambiguity-creating issues are highlighted by [84] such as scaling conflicts where values correspond to different scales, and confounding conflicts enabled by context-dependent properties. Various strategies have been implemented to address these issues, such as seeking further clarifications in question-answer systems [83], or utilising the PLIB system to tackle semantic heterogeneity in databases by employing universal identifiers to integrate data from diverse sources. In cases of concept fuzziness, fuzzy logic has been employed to convert imprecise concepts to precise [85] . However, in all these cases, the ambiguities are anticipated and acknowledged beforehand.

In the Social Sciences, ontology is not necessarily approached with the same level of formality. However, there is the concept of social ontology. Lawson, in his recent book [86], underscores the importance of studying social ontologies and notes that often, unexamined ontologies are accepted at face value, leading to the application of unsuitable methods. This is consistent with our observations about the ontology of bullying as discussed in this section. In the next section, I explore the use of the concept of bullying within a specific intervention program, aiming to get a deeper understanding of the degree to which ontology is clarified in practice.

4.7 Is there a consistent understanding of the term bullying in current interventions-Implications (Step 10)

The ambiguity of bullying does not only affect our communications. First, concept ambiguity suggests that there may not be a point in searching for a theory of bullying. Moreover, it raises questions about the already existing interventions. Bullying interventions have had mixed results [23–25]. For this reason, we explored the question: "Do intervention programmes employ consistent and concrete descriptions of bullying behaviours?".

I investigated the definition of bullying in three areas of the KIVA intervention program. First, I checked how bullying is defined in the background literature used to develop the program. Second, I looked at the interpretation of the program material, and third, the surveys and interviews to assess the program's effectiveness. Overall, I tracked seven evaluations. KIVA is listed under the Intervention Programs for Bullying [22] in the European Commission database. Also, among programmes with Evidence-Based Practices, it has been evaluated as a promising practice. It is worth noting that no programme has been assessed as a Best Practice.

The European Commission report [21] recognises the lack of an agreed bullying definition. One of the most common definitions used for bullying (in this or a similar format) is the one initiated by Olweus, who defines bullying as "the negative actions" of one or multiple persons towards one person, "repeated" over time, when the actor/s and the receptor of the behaviour have "asymmetric power relationship" [74]. A frequent bullying measurement method is the Bully Victim Questionnaire (BVQ) which uses the Olweus definition and a list of example behaviours: teasing, name-calling, making fun of, exclusion from groups, being ignored, physical attack methods such as hitting and shoving around, locking in rooms, harming the reputation of the victim [92]. The list is finalised with a general case "and other hurtful things like that" and the note that these behaviours do not count as bullying if they are done in a "friendly and playful way". Table 4.2 presents our findings.

This analysis of the KIVA antibullying program showed that existing intervention programmes do not sufficiently clarify the bullying concept. They operate primarily on the abstraction level of definitions and sometimes include examples of behaviour (see Figure 4.4). However, even in these cases, they leave much room for interpretation. Thus, teachers and other stakeholders are left to assess whether their observations match the intervention program's prescribed definitions and behaviours. The result is the implementation of programs in cases where their effectiveness has not been proven. But, given the concept's ambiguity, measurement methods are also called into question. We need to consider whether the concept's ambiguity allows for accurate measurement of the interventions. All these factors might explain why certain intervention programs appear to have mixed results. It could be that they have consistent results, but that measurement ambiguity does not allow consistent findings. It could also be that they are not consistently applied since stakeholders need to rely on their interpretations, which can be very diverse. We argue that we should make intervention plans (and models) that target concrete behaviours. Then we will be able to evaluate them and understand their true impact. Finally, we argue that similar problems may be present in intervention programs for other issues such as "sexual harassment".

Table 4.2: Evaluation of the KIVA program

Program/ EU Evaluation/ Sources	Bullying concept (Literature)	Bullying concept (Material)	Bullying concept (Measurement)
KIVA/ Promising/ [75, 87–91]	Olweus definition Additions to the definition: group character (victim, bystanders, bullies, supporters, and defenders) and the social character of bullying (social status) Keywords: aggression, violence Comments: No listing of bullying behaviours. to isolate	(Based on the available materials) Olweus definition Additions: bullying behaviours are intentionally harmful Comments: No examples of bullying behaviours Listing of non-bullying behaviours: arguments, disagreements, teasing or rough play	Olweus definition from the BVQ (Evaluation 1), Evaluation 2 adds a filter, namely the Solberg and Olweus criterion to characterise a behaviour bullying, Evaluation 3 included a simplified version for younger students, Evaluation 4 used the Revised BVQ with the addition of a description of bullying modes such as sexual, apart from the list of examples from the BVQ Evaluation 5 "presented ... a wide variety of situations in which bullying can happen" (the descriptions were not available for assessment) "to obtain unbiased estimates", Evaluation 6 builds on another evaluation, Evaluation 7 pointed that teachers do not always "recognise the key aspects of bullying" Measurement methods: Questionnaires (Peer reports, Self-reports)
Kiva evaluation	All Kiva materials use the Olweus definition of bullying. Since Kiva is a paid program, we do not have access to all the material but the available material list only a few examples of behaviours. Most evaluations use a more exhaustive list of behaviour examples, as presented in the BVQ. Evaluation 4 introduces new aspects of bullying not discussed in the first three evaluations, such as sexual bullying. Personal interpretations are allowed under the general term "Other hurtful things". Building on the observation from evaluation 7, we cannot be sure how bullying was presented to students and if measurements introduced news aspects not considered during implementation. Evaluations, with prominent examples the Evaluations 4 and 5, most probably will not count the same behaviours and thus might differ in their measuring capacity.		

4.8 The neurological basis of conceptualisation and miscommunication of knowledge holder insights

Figure 3.2 describes the creation of a social simulation model starting from data and theories. However, as we showed in Paper C, our modelling concepts, such as the concept of bullying, have implications for the quality and the validity of our results. Consequently, there is a need to start conceptualising the modelling process before the conceptual model stage. Figure 4.5 extends Figure 1.1 with Transition 0, and the "Experience and Mental Model Stage". These extensions express an updated understanding of the modelling process exemplified for the MARG model. The added features convey the importance of aligning the initial ontology of knowledge holder insights with their implementations in the conceptual and computer model descriptions.

The experiences are conceptualised in an implicit model of our brain, which we call the mental model. Then, experiences are shared with modellers or other investigators responsible for data collection and interpreted by their mental models. Figure 4.6 extracted from Paper D zooms into the stages from the perception of the experience until the start of the model creation process. For the context of the paper, we focused primarily on knowledge holders with first-hand experience. As a result, the blue side symbolises the person with the experience, and the orange side is the researcher conducting data collection.

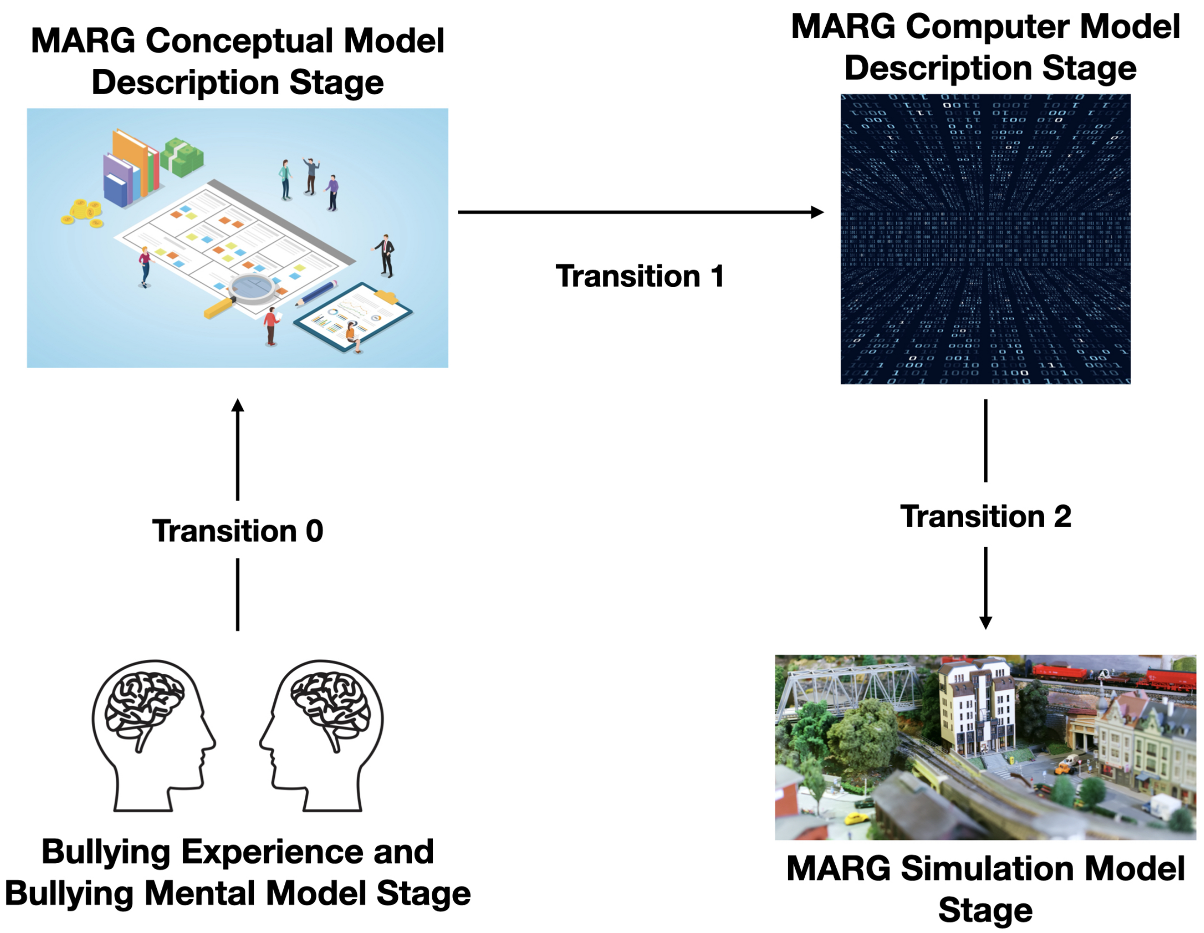


Figure 4.5: A Model of Modelling

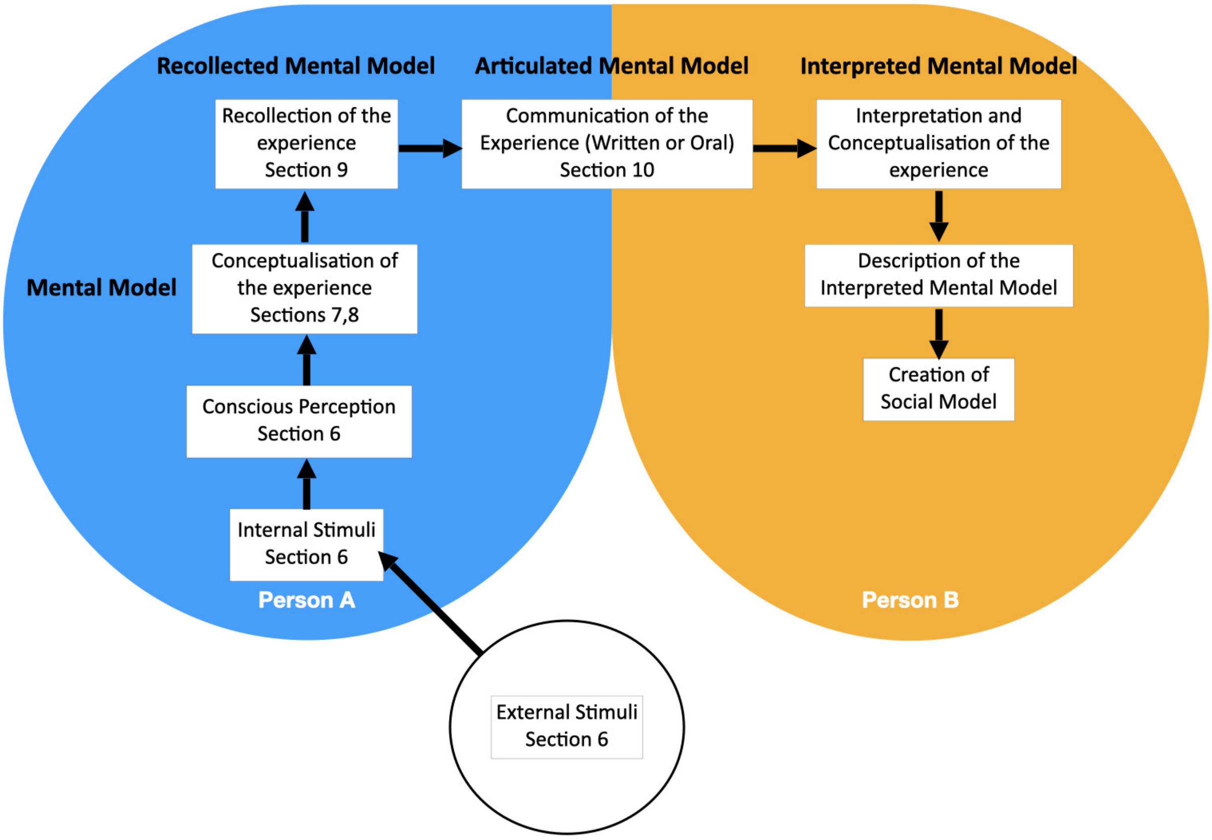


Figure 4.6: Data collection of human experiences: from the conception to the social simulation model drawn from Paper D

We wrote Paper D to present the errors or issues that enter into the model creation process from how we conceptualise and communicate our experiences. To accomplish this task, we created a conceptual model for the conceptualisation and communication of the knowledge holder insights using literature and combining sources from different disciplines. Figure 4.7 illustrates the emergence of both conscious and unconscious perception. Figure 4.8 depicts the process of forming knowledge holder insights and their subsequent communication. In both figures, the elements marked in red represent potential sources of inaccuracies in our perception formation, experience conceptualisation, and the potential for misinterpretation of our knowledge insights. Except for the first item, the following list summarises the main findings presented in Paper D. For a comprehensive analysis and detailed source references regarding the content, please refer to Paper D.

- **Brain Basics and Stimuli** To better explain the content of Paper D, I refer to the brain structure and function. One of the brain's main components is the neural network. Neurons are nerve cells and their connections are called synapses. Not all neurons are connected. Most of the synapses are formed in the early stages of our life and strengthened or weakened later. Each synapsis plays a role in the activation of the attached neurons upon external or internal stimuli . On the level of the brain, stimuli become an electrical or chemical signal. Depending on their use, some connections become enlarged while others diminish in size and are not activated with signals.

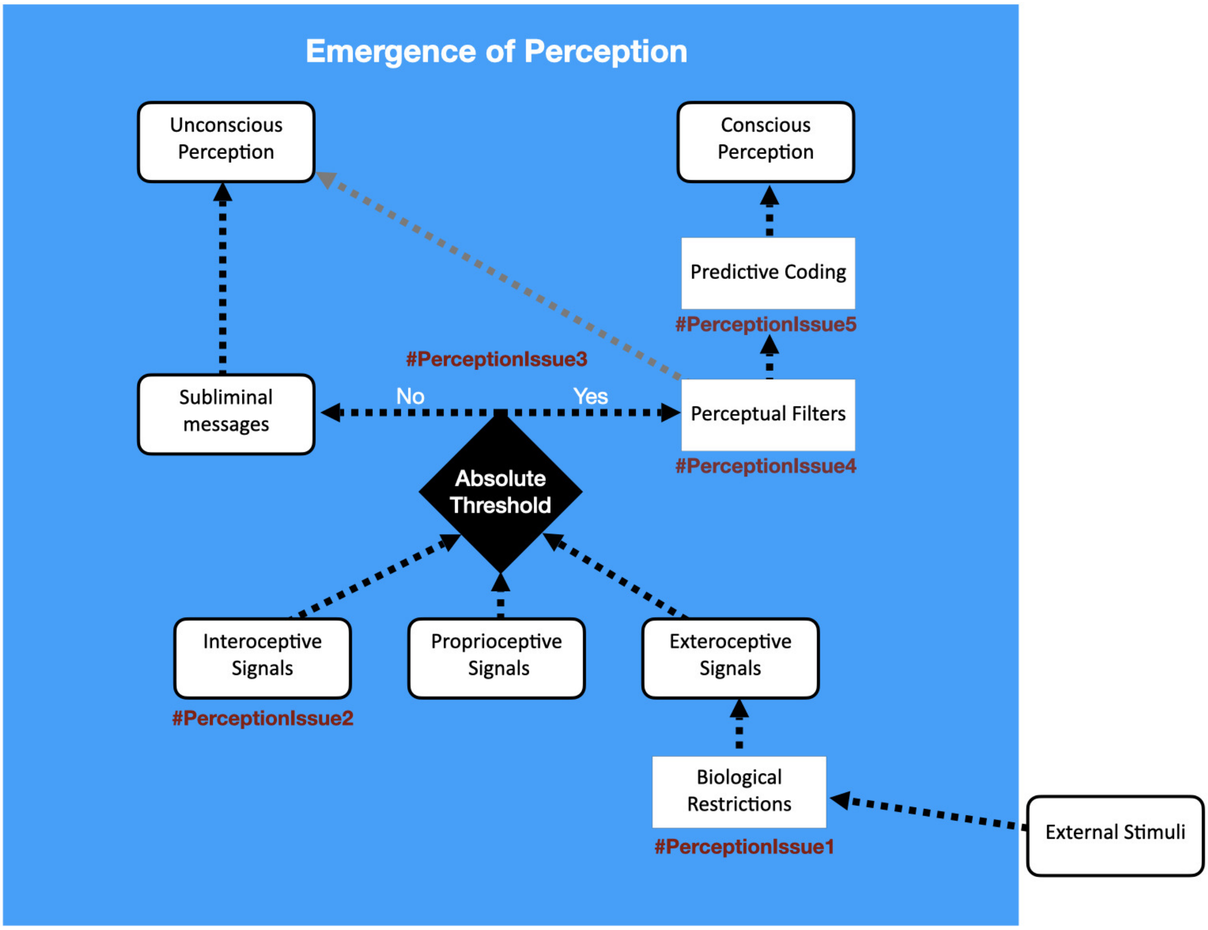


Figure 4.7: Model of Conscious Perception

- **Emergence of Perception** Understanding perception is vital as it is closely intertwined with the process of conceptualisation. As depicted in Figure 4.7, we differentiate between three types of signals: exteroceptive, interoceptive, and proprioceptive. Exteroceptive cues refer to external stimuli perceived by sensory information, the perception of which is constrained by our biological capabilities (#PerceptionIssue1 in Figure 4.7). On the other hand, interoceptive and proprioceptive signals are internal. Interoceptive cues provide information about the sensations from our internal organs, with the level of detail depending on the specific organ (#PerceptionIssue2 in Figure 4.7). Proprioceptive signals, meanwhile, offer insights into our body's position and activity. For a signal to influence our perception, it must surpass a certain energy level, known as the absolute threshold (#PerceptionIssue3 in Figure 4.7). However, even if this threshold is met, the signal may not necessarily reach conscious perception. Perceptual filters, such as sensory adaptation and selective attention, further limit the number of signals that enter our perception (#PerceptionIssue4 in Figure 4.7). Signals that are not consciously perceived can still contribute to our unconscious perception. Ultimately, our understanding of our environment is shaped not only by the actual signals that manage to overcome these barriers but also by our previous experiences through a process known as predictive coding (#PerceptionIssue5 in Figure 4.7).

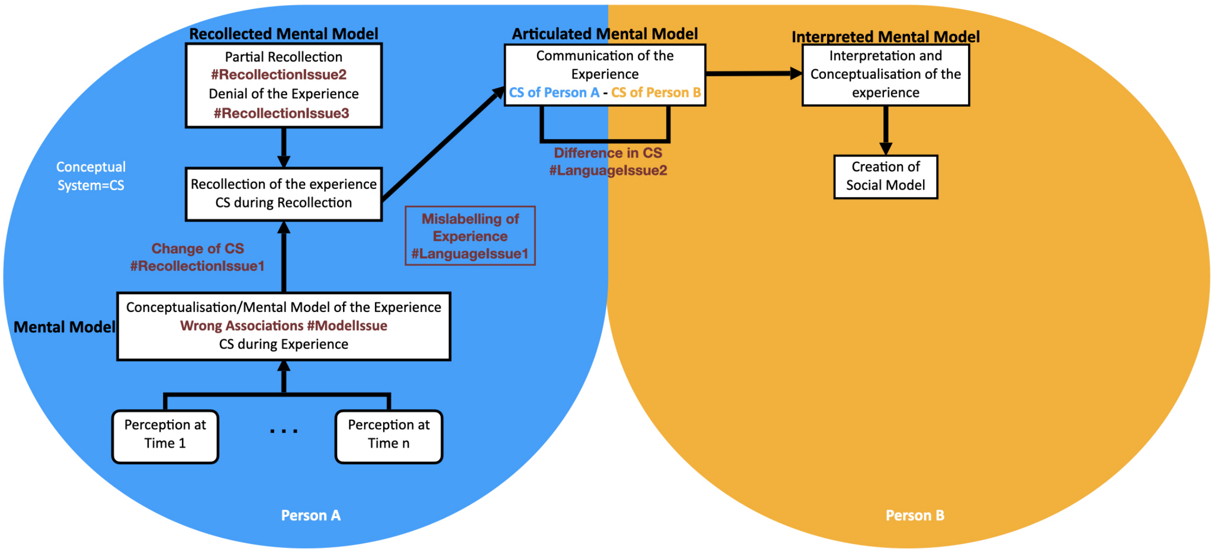


Figure 4.8: Sources of errors in conceptualisation and communication of human experiences

- **Concepts, Predictions, and Mental Models** As mentioned in Paper D drawn from [93], "Details of different experiences with a perceived similarity become grouped in neurons. This multisensory summary is an abstraction of the similarities among the experiences and corresponds to a concept". Contrary to older beliefs about concepts, concepts are not static but are very dynamic and highly dependent on context. We employ our conceptual system to classify our present experiences. Amid an experience, our perceptual cues not only activate the neural systems corresponding to the situational elements but also the associative areas. This empowers the brain to conduct simulations using the activated neural networks, facilitating predictive perception and appropriate actions based on our situation. The way we group and associate, our conceptual system, holds information and influences our perception and future actions. In Paper D, we established a correlation between the conceptual system and mental models, advocating for a bridge between the disparate bodies of literature.

- **Errors during Conceptualisation** Our associations are potentially fallible, leading to the possibility of erroneous mental models (#ModelIssue in Figure 4.8). Surprising outcomes can serve as catalysts for updating these mental models. Instances of such flawed associations can be seen in superstitious beliefs. An example of the origin of erroneous associations is showcased with cognitive biases such as confirmation bias.

- **Recollection** An experience is often shared after it has been conceptualised, necessitating its recollection. However, as the literature on mental models underscores, there is often a discrepancy between people's actions and their accounts of those actions. We have pinpointed several issues related to recollection, including conceptual changes (#RecollectionIssue1 in Figure 4.8) , incomplete recall of behaviour (#Recollection Issue2 in Figure 4.8) , and denial (#RecollectionIssue3 in Figure 4.8) . For instance, weak connections might fade, implying that certain details, particularly those deemed insignificant, may no longer be associated with a specific experience.

A prime example of conceptual change is memory distortion, which can manifest as the reconstruction of experiences with inaccurate details, as seen in cases of suggestibility. Incomplete recollection can stem from various sources. Examples include the inability to recall in the absence of cues or failure to consolidate episodic memory. Lastly, it's possible for an individual to deny an experience, as often occurs in cases of trauma.

- **Communication of Insights and mistranslation** Concepts and words are separate entities. Mental models are formulated using concepts and articulated through words. Each word is a concept in itself and can signify one or more concepts. However, not all concepts can be fully expressed by words. Concepts are dynamic and can differ among individuals and cultural groups. For instance, an owl is perceived as wise in many cultures, but as inauspicious in Persian culture, indicating diverse conceptual development. A single word can denote entirely different concepts, as seen with the example of concept ambiguity in Section 4.6. However, words like "bullying" can have different conceptual interpretations for different individuals. Overall, it's possible for two people in a conversation to interpret the same word differently (#LanguageIssue2 in Figure 4.8). Communication errors can also occur, even with a reasonably accurate perception, when we mislabel our experiences (#LanguageIssue1 in Figure 4.8).

The findings of Paper D hold several implications for our modelling process. First, we must align our concepts when conducting data collection and throughout the modelling process. Methods of mental model elicitation aid towards this direction (see [94, 95]). Based on our analysis with Paper C, we also propose the use of concepts of concrete actions as they provide more clarity. For instance, a concept such as "run towards target A" offers better clarity than a concept such as "work out". We believe it is the same with concepts describing social conflict.

Paper D also sheds light on our limitations regarding models of human behaviours. In the best-case scenario, we can extract an accurate mental model related to an experience such as bullying. The realisation that a mental model of an experience is actually based on the neural network of our brain implies that future progress in neuroscience might allow us to increase the accuracy of the elicitation methods. However, it is not easy to improve another's recollection or understand how correct their inferences are. In addition, it is almost impossible to clear out perceptual issues of past experiences. However, we should train ourselves and others to question whether our mental model is correct and to look for more cues in future experiences. We can be very aware of these issues and relax our confidence in our results. We can also include them as possible risks for model inaccuracies. Finally, I argue that we should start describing our modelling process from the conception of human experiences to encapsulate errors in models and potential solutions fully.

Chapter 5

Developing the DSM tool

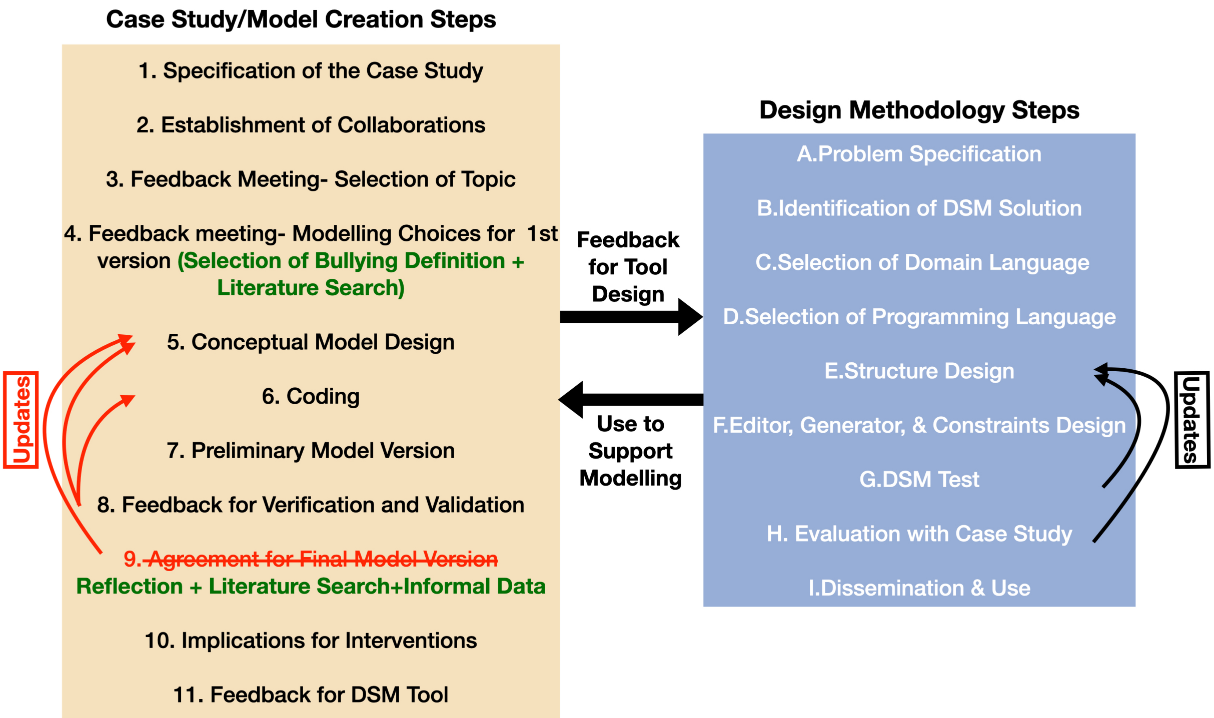


Figure 5.1: Applied Methodological Steps: Red indicates unfinished planned steps, while green represents unplanned but realised steps

5.1 Applied Methods

Figure 5.1 shows the implemented steps. As you will notice, some of the planned steps were not performed. Figure 5.1 also captures where the literature search, reflection, and data collection phases were actualised in the project timeline. The differences between Figure 3.3 and Figure 5.1 are marked with red and green colours. One thing to remember about Figure 5.1 is that it does not display the actual coordination between the two parallel development processes (yellow box and blue box). To remedy this shortcoming, the sections of this chapter are sequenced in an order that resonates with the actualised time plan. As a result, the chapter

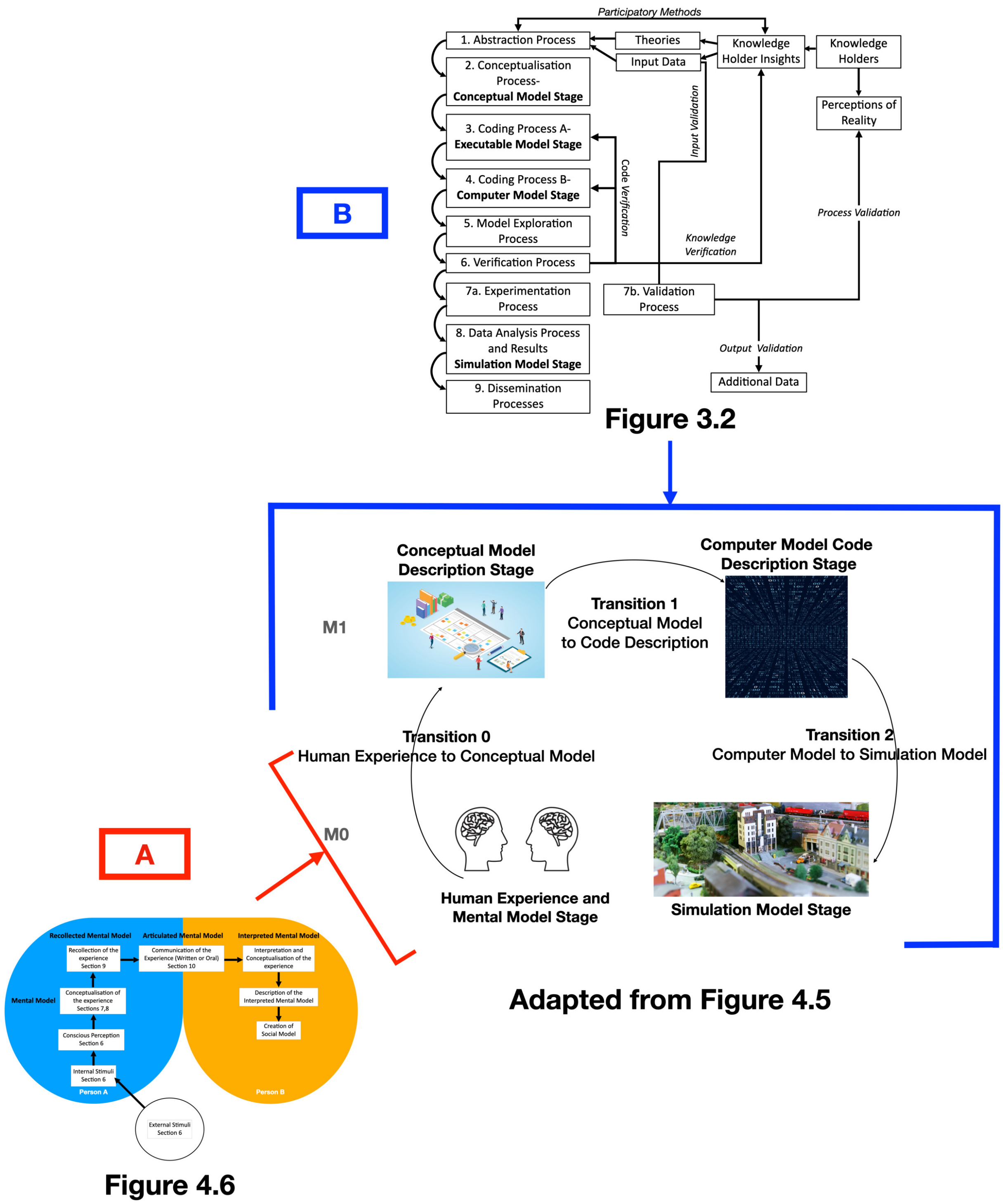


Figure 5.2: A knowledge map for Figures 4.5, 4.6, and 3.2

starts with the model creation stages and continues with the progress of the tool creation. That said, you will need additional information to understand the project's progress completely. Even though steps A and B come after the model creation process (Steps 1-10), a form of the problem was acknowledged from the start. Aside from the problem, a tentative DSM solution was also selected in the early phases of this research. The complete problem is presented as Step A, but some parts were revealed after Step H.

5.2 Problem exploration (Step A)

Figure 5.2 maps the knowledge presented so far in the thesis in order to serve as a map. The centre of Figure 5.2 corresponds to an adapted Figure 4.5. Area A of Figure 5.2 is drawn from Figure 4.6. Area A connects the modelling process to reality. It details how the human experience becomes a mental model and enters the data collection phase, which corresponds to Transition 0. Area B of Figure 5.2 corresponds to Figure 3.2 and shows a detailed sequence of the processes and stages from Transition 0 to the Simulation Model Stage. From this point onwards, I will refer to the Areas A, B and Centre of Figure 5.2 instead of the included figures from the previous chapters. Step 1 in Figure 5.2 (Area B) captures part of Transition 0 with the "Abstraction Process". Step 2 of Figure 5.2 (Area B) corresponds to the rest of Transition 0 processes. Transition 1 of Figure 5.2 (Area A) summarises the "Coding Processes" (Steps 3 and 4 of Figure 5.2 (Area B)). Processes 5, 7 and 8 of Figure 5.2 (Area B) yield the "Simulation Model Stage", which corresponds to Transition 2 of Figure 5.2 (Area A). The Verification processes (Process 6 including the arrows Code and Knowledge Verification of Figure 5.2 Area B) check aspects of Transition 0 and 1 but typically occur after the transitions are completed. In total, we identify five model stages: the mental, the conceptual, the approximation, the computer, and the simulation model stage.

The process of Transition 0, in the context of human behaviour and conflict resolution models, involves integrating insights from human experiences into the conceptual model. This process begins with knowledge holders sharing their insights, which have become part of their mental models, with researchers on the modelling team. The researchers use their own mental models to conceptualise the communicated insights and express them in the conceptual model. Data collection methods such as interviews, surveys, transcripts, and published work are common ways of acquiring knowledge holder insights. In participatory approaches, such as in the case of the bullying model, insights may also be gathered during feedback and consultation meetings.

Knowledge insights may be qualitative or quantitative. It is easier to represent quantitative insights into conceptual models than qualitative insights because of their numeric format.

Regardless of their qualitative or quantitative character, knowledge insights vary in levels of completeness. For instance, published work may provide comprehensive theories or conceptual models. In contrast, insights obtained through data collection methods may only offer a glimpse of a potential theory. Even if the published conceptual models need adjustment to fit the ABM framework, complete theories leave less room for interpretation.

In our case, the academic community did not provide a comprehensive theory or conceptual model on bullying but offered only scattered pieces of qualitative information. To fully understand and analyse the available input, we developed our own theory and expressed the received insights in a new conceptual model. For instance, we expressed the definition of bullying using the theory from Thibaut and Kelley [78] as our base. Exclusion was expressed in relation to the negativity of the interaction. While this enhanced our understanding of the topic, it may have diverged from the original intentions of Schott and Søndergaard's definition [73], which counts as a misalignment of knowledge holder insights. Overall the less complete and more qualitative the insights, the more likely the modelling team is to misinterpret and misrepresent them.

Misalignment of insights can also occur during Transition 1 (see Centre of Figure 5.2). To better understand the errors during this stage and the areas of interest related to these errors, the following sections are dedicated to the explanation of a new modelling framework.

5.2.1 The OMG Hierarchy

The Object Management Group (OMG) has proposed a conceptual framework for models that consists of four layers [96]. According to this framework, the bottom layer, denoted as M0, contains all systems, including real-life "objects". The next layer, M1, contains system descriptions, and the top layer, M2, includes the languages used to create these descriptions. The fourth layer is not relevant to our work. We discussed the various layers in detail in our paper [96], and I will use this paper for the following analysis.

A system is created when we apply a specific perspective to reality. The perspective influenced by our motives highlights certain aspects of reality while downplaying others. To understand this concept, consider a particular location and observe it from different perspectives. You might consider the different species of flora and fauna and their interactions when viewing the location from an ecological perspective. You could focus on how humans have interacted with the land and changed the morphology of the area. If you view the same location from a painter's perspective, you might focus on the colours you see, the lighting, the noise, and the energy. These different perspectives allow two people to perceive different systems based on the same reality.

A model is a system that refers to another system but is reduced compared to the referent system. According to the definition of a system provided in [96], a system (which includes a model) is "a changing set of executing objects and their properties. These objects interact with each other and entities in the system's environment, resulting in changes in the objects and properties. Objects may be existing entities like devices, and they may be entities that have to be made as part of the systems development. This way, a system is a set of possible executions, i.e. a set of object configurations that exist at different time points".

Model descriptions are not parts of the model. Instead, they are separate entities. Consider the MARG model as an example to clarify the distinction between model and model description. According to the framework proposed in [96], the MARG simulation model includes all possible executions achieved through simulations. The MARG model is not the computer code written in a programming language, which is just the model description. Nor is it the ODD protocol, which only describes certain aspects of the conceptual model. Moreover, the MARG model is not the output data analysis, even if the model has been fully explored. Instead, the MARG model is the actual behaviour described in the complete result analysis. A final note on model descriptions is that some are also used as model prescriptions. In other words, some descriptions help us create models.

Finally, the language we use to create model descriptions is located at the M2 level. The language is important because it defines the semantics of the written text. As mentioned in Section 2, formality is the property of having explicitly stated semantics. A formal language at the M2 level means that the model description has the potential to be formal. For example, the language of mathematics is formal, and therefore, mathematical descriptions are formal. Whereas ambiguities in concepts can occur in informal models, such as those we encountered with the term "bullying", ambiguities in formal descriptions can arise primarily when they are incomplete. A consequence of the previous analysis is that there are no formal models, only formal descriptions, as formality pertains to language statements.

5.2.2 Model Context and Model Physics

In Paper E, we introduce the idea that a model can be divided into two components: "model physics" and "model context". These components are considered M0 objects within the OMG hierarchy, meaning they are parts of the model.

To understand what I mean by "model context", think about the system defined as "the set of all possible executions" involving changing objects. This idea assigns the system a dynamic nature, like a living organism. If we take a snapshot, we get a static view, one of the possible modes of existence for the system. The model context is the way we link this snapshot to reality. It is the details that set the frame for the model. For instance, in our MARG model, we could have a snapshot from the simulations where one student is having a negative interaction in a supervised environment (denoted as forced-interactions in the model descriptions). The model context would be the placement of this scene in a FYSE study group environment. The forced interaction would be further embellished with information about the type of group work that brought the two students together, perhaps the course itself. We model university students in MARG. Thus, we would envision the students as adults. And we can continue like this to specify the model context. In the end, we would be able to match this context with real-life experiences more efficiently. Based on my comprehension of the field of Knowledge Representation, the context of a model is tied to the ontological commitments we undertake, which constitutes one of the key facets of Knowledge Representation [8].

Model physics is what drives one snapshot to the other. Model physics is about the "physical" mechanisms in the model. At its core, as we mention in Paper E, model physics is all the potential calculations structured in sequences. Model physics is expressed in the programmatic formulation of the model, allowing us to experiment with the model context and come up with different outcomes. However, model physics is not the programmatic formulation in the same way a mathematical function (M0) is not the mathematical formulation of the function (M1). From the Knowledge Representation literature, model physics is connected to the component of logic [7].

5.2.3 Improved understanding of the model stages

The consequences of the two frameworks become clear in this Section when we combine the information. Figure 5.3, adapted from Paper E, provides a graphical understanding of the outcome to support the conclusions.

- **The model creation process starts with different mental models and produces a simulation model. The intermediate transitions are actualised using model descriptions.**

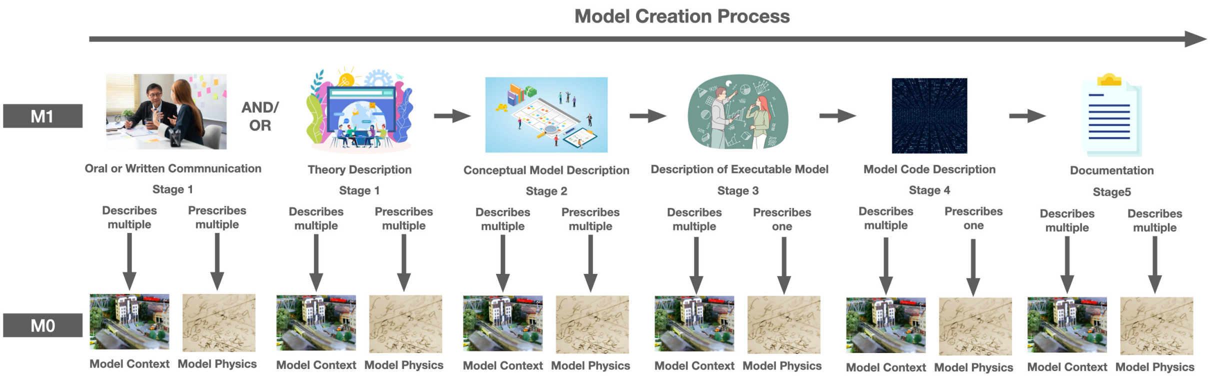


Figure 5.3: Model transitions during model creation (adapted from Paper E)

- **Specifying model physics is one of the main goals of the model creation process.**

  The model creation process is driven by the intention to create a social simulation model. Model physics is part of the simulation model and is what allows us to explore the simulation model using computers. Because of its significance in the simulations, one of the most important targets of the complete social simulation process is to specify model physics.

- **Model Physics are fully defined in the Executable and Computer Model Descriptions.**

  From the executable model description onwards, the model physics is fixed. By fixed, I mean that the model descriptions include a precise programmatic formulation that prescribes only one model physics. This is portrayed in Figure 5.3 with a link showing that the executable and computer code model descriptions prescribe one model physics. In contrast, in the previous stages, the model descriptions prescribed more than one model physics. This phenomenon has been observed by other researchers who note that theories are interpreted dissimilarly by different modellers [97], meaning the same theory can yield multiple computer model descriptions [98]. Since the core component of the code is the programmatic formulation, using our framework these results mean that one theory has ambiguous or incomplete programmatic formulations. Figure 5.3 portrays this with the one-to-many prescription relationship between theory description and model physics.

- **Model descriptions do not prescribe but only describe the model context, which is an important part of our mental models.**

  The model context description (M1) provides the required information to match a model with real-life events (M0). However, even if two people observe the same event, their mental models may lead them to interpret the stimuli and conceptualise the event differently. This is due to the unique way our brains store and structure information based on our individual conceptual systems, leading to different associations and understandings of the same real-life objects (as explained in Paper D). We make our model descriptions specific to enable stakeholders to effectively apply the insights gained from the model. However, they must be general enough to capture all potential conceptualisations of the same event. As a result, each model description should naturally describe more than one model context. In contrast to model physics, the model creation process does not generate a very concrete model context description because this would reduce the relevance of the model. Moreover, since the model context will always be richer in our brains, we are not trying to create it but describe it. Thus, the model description and model context do not have a prescription relationship.

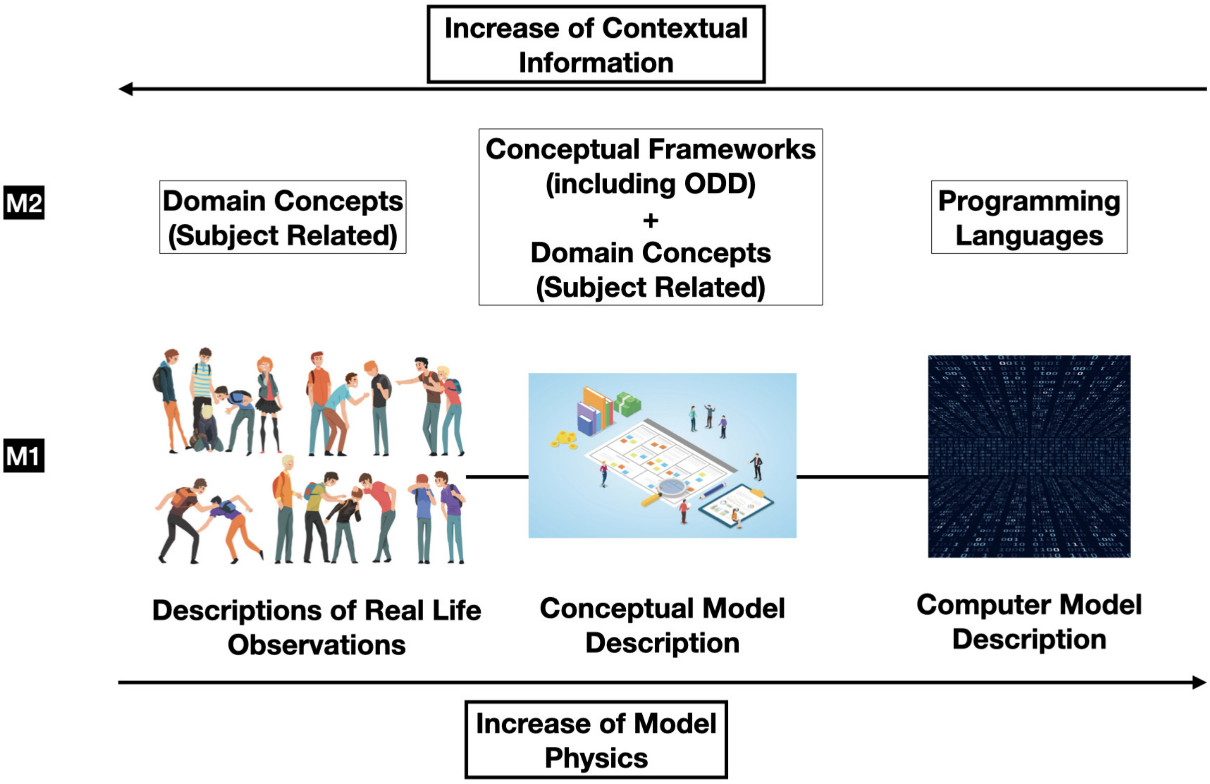


Figure 5.4: Languages for different Model Descriptions, extracted from Paper E

- **The model creation process acts as a funnel reducing the number of potentially prescribed models as we progress.**

  In the early stages of model creation, the physics of the model is not well-defined. It becomes clearly outlined as the process progresses (see Model Physics in Figure 5.4). At the same time, the context in which the model operates is rich and detailed, but as the model creation process moves forward, this context is described less explicitly (see Contextual Information in Figure 5.4). However, despite this lack of explicit description, the model context is still retained in later stages due to the connection with the earlier stages. This retention, combined with the crystallisation of model physics, helps to decrease the range of potential prescribed models. The ultimate goal of model creation is to define model physics in a way that allows for model exploration through simulations. This intention requires the narrowing down of the range of potential models. But it's important to note that there may still be a range of models even at the final stage, as the model context is inherently subjective and can lead to multiple models with the same underlying physics.

- **Descriptions in different model stages are expressed using different languages.**

  The transitions between stages in a model are driven by a change in the language used to describe it (see Figure 5.4). The shift from the conceptual model description to the computer model description is enabled by the use of programming languages instead of conceptual frameworks. Formal languages provide explicit semantics and the possibility for formal description, which allows for the expression of model physics.

5.2.4 Challenges and Errors

To fully understand how misalignment emerges, we should consider the sources of errors in the different stages.

- **Errors during Transition 0:** Figure 5.4 shows the change of language from the description of experiences and observations to the conceptual model description. Descriptions in natural languages, such as the knowledge holder insights, must be "translated" to fit common conceptual frameworks. Conceptual Frameworks (M2), such as the ODD protocol for conceptual model descriptions (M1), are more formal in comparison to natural languages (M2). As a result, the translation process may require the use of assumptions and adjustments to bridge the gap between the two languages. However, these assumptions and adjustments are not necessarily neutral. They could unintentionally alter the content of the knowledge holder insights. Note that the conceptual model descriptions are considered still informal [99]. Incomplete descriptions and mismatched concepts between the knowledge holders and the modelling team add to this challenge (see Paper D for references in these issues).

- **Errors during Transition 1:** Similarly to Transition 0, Transition 1 requires a translation from one language to another. This time we translate the semi-formal conceptual model descriptions to the formal computer model descriptions. Unlike programming languages, conceptual frameworks allow more abstract and incomplete descriptions. In our framework, this is visible with the one-to-many relationship between conceptual model descriptions and model physics and the one-to-one relationship between computer code descriptions and model physics (see Figure 5.3). Again, the modelling team must make assumptions to fill the gaps. Galan et al. refer to two assumption categories: accessory and core assumptions, with core assumptions playing a significant role in the model behaviour and accessory having a minor impact [31]. The introduction of assumptions may alter the implementation of the insights, and, in some cases, this causes misalignment. The misalignment risk is increased with coding errors, bugs and the limitations in the expressivity of programming languages [31]. Using our framework, the misalignment implies that the computer model description might refer to model physics and model context that are not allowed in the previous modelling stage.

- **Errors during Transition 2:** Transition 2, as defined in the central part of Figure 5.2, contains all processes that help us conceptualise the Simulation model. These include model exploration, model experimentation, simulation data analysis, the attribution of model outcomes to parts of the model description, and model documentation. The implementation of insights has already been finalised in the computer model description. As a result, most of the errors associated with Transition 2 are not of interest to the scope of this dissertation. The only exception is the documentation process. Model documentations describe all model aspects, including the implemented insights. They are informal descriptions, and to my knowledge, there are no formal checks of their correctness. Therefore, model components may be misrepresented in the model documentation. Hence, when knowledge holders read the model documentation, they may fail to identify and solve implementation misalignments.

Based on the problem exploration, I present an updated problem statement:

> **Updated Problem statement: What is an effective method to retain knowledge holder insights, which correspond to parts of model physics and context, during the model creation process so that the conceptual and computer model description accurately represent them, resulting in a simulation model of good quality in terms of insight representation and potentially valid advice for conflict resolution?**

As I mentioned in Section 1.1, the social simulation methodology requires collaboration with people who do not necessarily have programming skills and cannot follow the trail of their insights in the model creation process. This is the root of the stated problem. To understand the significance of this issue, in a study of 2013 [100], Annie Waldherr and Nanda Wijermans collected different categories of sceptical responses towards social simulation. Although the reactions address specific aspects of presented social simulation models, such as model assumptions, model complexity, and model simplicity, the authors believe that there are two main drivers behind the scepticism. The first driver is the incomprehensibility of social simulation models by knowledge holders such as social scientists. In other words, the study identifies the root of the rejection to be the same as the root of the Problem Statement and confirms the severity of the problem.

The second identified driver is that sceptics want to adhere to their familiar methods [100]. My observations from collaborations with different academic knowledge holders, such as social scientists, psychologists and archaeologists, also confirm the conclusions of Waldherr and Wijermans's study, especially regarding people who focus on qualitative methods. Within social sciences exists a divide [101] referred to by Edmonds [102] as the "Qual-Quant Wars". Namely, some social scientists favour quantitative methods while others strongly prefer qualitative methods, resulting in an insufficient methodological mix [103]. Quantitative data and insights offer fewer challenges when translated into ABM components. Therefore, it is imperative to understand how qualitative knowledge-holder insights are implemented.

5.2.5 Existing Methods to Approach the Stated Problem

This section summarises existing ways to approach the stated problem. The identified methods are presented through examples and not a complete literature review. Due to the lack of a consistent expression for the indicated problem, when searching for relevant publications, I used various keywords such as "verification", "validation" and "participatory methods". I drew upon my knowledge of the social simulation literature to pinpoint methods that have contributed to solutions for the targeted problem, regardless of whether it was their intended purpose.

- **Transition 0** The goal of Transition 0 is to express the accumulated knowledge insights with the ABM framework resulting in a conceptual model description.

  Researchers have used different methods to gather the required qualitative knowledge from perspective knowledge holders. Examples include conceptual maps [104], content analysis and cognitive maps as written in Paper D. Sometimes more than one researcher is involved in the data collection to crosscheck their notes (see [105]). These methods have been called knowledge engineering, knowledge elicitation, or mental model elicitation methods [59, 94, 95].

Following the knowledge acquisition, the information must be assembled into one conceptual model. Different methods have been applied to synthesise information from the mental model elicitation process. An example is pattern recognition, as noted in Paper D. Formatting the information in conceptual frameworks, such as the ABM framework, is a challenging process [106]. In addition, the process is often not well documented [102, 107]. New frameworks such as MAIA [108] have addressed this issue by introducing the conceptual format during data collection. As a result, the data can be used directly or with minor adjustments to fit the conceptual framework. Other examples include the KNETS framework that links collected data directly to agent rules [109], the framework for narrative analysis to identify the mental model context and partially match narrative aspects to ABM components [110], and the frameworks to guide knowledge elicitation mentioned in [111].

Aside from the use of data, the use of theories in conceptual model descriptions is not well structured and leaves a lot of room for interpretation, as shown in [97, 98]. However, there have been systematic efforts to assimilate and format theories, either with the use of frameworks or by targeting a specific domain and synthesising existing theories (see [97, 112].

On a final note, researchers have combined games and ABMs in participatory approaches. The similarity of ABM simulations to gaming experiences [55] allows researchers to collect data from participants [113]. As Szczepanska et al. write in some cases, the ABM is merged in the game so that data integration is not necessary [113]. They continue to report that games have also been used to allow the co-creation of the model. This is possible because, as Barretau et al. mention, playing the game enables participants to experience the model [55].

- **Transition 1** As discussed in previous chapters, we aim to rectify the coding errors with the verification processes. Code Verification revolves around the correction of "coding bugs" or, in other words, programming code that has an unintended function in the model. This can be done with processes such as reading the code or linking code to model behaviour [58]. Knowledge verification, which some researchers call validation, face validation, or ontological validation [58–60], involves active checks of the simulation model by knowledge holders. To perform knowledge verification, knowledge holders must have access to the model. The COMSES website provides a database of models open to everyone to increase model code accessibility [114]. However, as Tubaro and Casilli note, academic knowledge holders may not be equipped with computer skills [103]. Researchers have additionally worked on increasing model literacy with [115], a first step to equip domain experts with the necessary skills to understand models. In addition, different methodologies have been incorporated to increase model readability and transparency. This is accomplished by replacing the model code with a model description in a familiar language. For instance, in some cases, researchers have used simplified diagrams capturing the conceptual model [116]. Formal ontologies are perceived as a tool to reveal and clarify the contents of a model and its potentially ambiguous elements [84, 99]. They are more comprehensible than programming code, yet maintain a level of formality. Some researchers propose the use of diagrams exposing ontologies by describing the relationship of modelling components to improve model transparency [117]. Other methods to allow knowledge holders to understand the simulation model include model visualisation techniques or model documentation [58, 118]. Examples of documentation techniques include the ODD protocol [119] and the Rigour and Transparency Reporting Standard [120]. Afterwards, knowledge holders can improve the existing implementation of insights through communication, such as in the case of focus groups [121].

Finally, a promising approach is presented in [122, 123]. This approach addresses the errors from Transition 0 and 1 for simulation models that support policymakers. It includes the transformation of narrative text to conceptual description using a conceptual framework called CCD. CCD also acts as a metamodel that semi-automatically generates computer model code. The resulting code keeps traces that connect coding parts to the initial textual input. Overall, this approach allows increased model transparency and links the original input to the implementation. However, it is limited to knowledge-holder insights expressed in policy scenarios.

5.3 Proposed Solution (Step B)

Based on Section 5.2.4, the causes of misalignment can be traced to incomplete descriptions (Issue 1), modelling assumptions (Issue 2), and coding bugs (Issue 3). The modelling team checks the coding bugs during the Code Verification phase. With Knowledge Verification, knowledge holders evaluate the implementation of insights. Ideally, this means they assess whether the introduced assumptions alter the content of their insights. However, their evaluations are limited. This is because they access the model code through potentially inaccurate model descriptions (Issue 4).

Figure 5.5 summarises the DSM framework, the current situation and the identified DSM solution. Paper E presents a large part of the solution, while Paper A details our methodology.

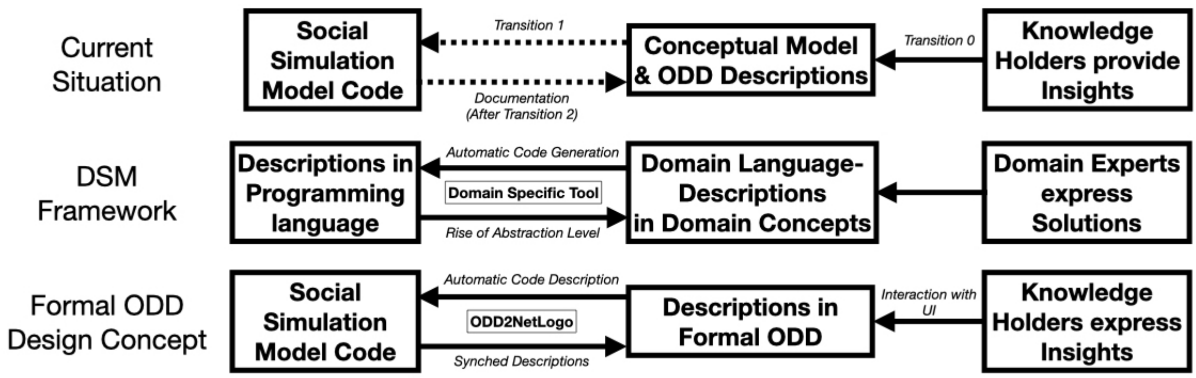


Figure 5.5: Proposed Solution in the DSM framework adapted from (Paper E)

Our solution, shown in Figure 5.5, includes three main components :

- The Formal ODD design concept. If we place the ODD protocol in the position of the domain language of the DSM framework and the social simulation model code in the position of the programming language, then we get the Formal ODD design concept (see Figure 5.5). Similarly to the current practice, knowledge holders can discuss modelling choices on the conceptual level and express their insights in the conceptual model description. However, since the model code is automatically generated from this description, the expressed insights are accurately portrayed in the computer model description (Resolution of Issues 2 and 3). Aside from the resolution of issues, the solution bypasses the tedious coding processes and reduces (or eradicates) the need for Verification checks.

  This idea is an improved version of existing proposals for solutions, which include the use of abstractions and code for Knowledge verification [31] and the combination of the ODD protocol and code for improved model documentation [124]. For previous work related to this type of solution and especially the use of meta-models, please refer to Paper A.

- The Formal ODD domain language. Currently, ODD descriptions can be vague and incomplete. We formalised and extended the ODD language to utilise the ODD concepts in the DSM tool. Since the language is formal, the conceptual model description, including the expressions of knowledge holder insights, are formal descriptions and must be complete (Resolution of Issue 1). As highlighted in [125], formal languages facilitate a comprehensive description of a system while they can maintain a level of abstraction higher level than that of programming code. At the same time, the ODD description can act as model documentation eliminating the time-consuming documentation process. Whereas documentation may introduce misrepresentations of model components, the Formal ODD description for the conceptual model description is always synched with the model code because it also works as a model prescription (Resolution of Issue 4).

- The ODD2NetLogo implementation. Our design choices and methodology formed the ODD2NetLogo DSM tool. The tool implements the Formal ODD concept and integrates the Formal ODD language.

5.4 Design Choices (Steps C to G)

The Formal ODD domain language is based on the ODD protocol for several reasons (Step C of the Methodology Figure 5.1). First, the ODD protocol is primarily used to document social simulation models [119], which means it is suitable for both documentation and conceptual model description. Second, it is widely accepted in the Social Simulation community. For instance, the JASSS journal recommends the usage of the ODD protocol when publishing model-related work [126]. Another point is that most social simulation researchers are familiar with the ODD protocol. Moreover, even if it is not formal, the ODD structure is well-defined and facilitates formalisation. Finally, the protocol offers a state-of-the-art means of documentation, fit for model replication, since it is regularly updated (see, for instance, [119,127,128]).

We chose the NetLogo language as the DSM programming language (Step D of the Methodology Figure 5.1), in which the models are automatically generated. The reasons behind the choice of NetLogo are recorded in Paper A and include its popularity, the fact that it is freely provided to everyone, its ability to fit an extensive range of models and, particularly, its ease of use and inclusion of a toolkit [129]. The NetLogo simulation platform is used to perform the processes of model exploration, and model experimentation [130]. The visualisation aspect of the platform makes it user-friendly. Alongside the DSL tool, it allows knowledge holders without familiarity with modelling or programming to immerse themselves in the simulation experience and better understand the model creation outcome. Its primary disadvantage is its reduced capacity to code simulation models, especially in comparison to general-purpose programming languages.

The DSM tool was developed in the MPS platform [131]. The MPS platform helps the implementation of compilers and languages by offering a framework for language development in which we define specific language elements to create the language. The fundamental elements are the structure, the editor, which acts as a user interface, the constraints and the generation rules. MPS facilitates language implementation and integrates the language components. A significant advantage of using MPS is that we do not need to define additional abstraction rules to generate the model description from the code, as it is part of the platform function. The person who wants to use the DSM tool only needs to access the user interface of the language. For our tool, which we named ODD2NetLogo, the user creates a new ODD, inputs a model description and presses a button to generate the text with the model code (for screenshots of the ODD2NetLogo, see Paper E).

The process to create the ODD2NetLogo DSM tool is analysed in Paper A. A summary of the methodology is provided in the remaining of this Section. We began the design process with the Formal ODD language structure. To define the language structure, we extracted the ODD concepts from the ODD protocol and complimented them with NetLogo concepts to enable code generation. Then we proceeded with the first editor version. The editor serves as a user interface. We designed the editor to match the appearance of an ODD template in terms of the required information. We added additional questions to formalise the ODD protocol. An additional editor requirement was to allow only structured input. We met this criterion by designing the user interface as text with gaps for user specifications. Some of the gaps contain menus to guide the user input. Next, we supplemented the restrictive character of the editor using constraints that signalled an error to the user when the information was unsuitable. Finally, to avoid inconsistencies in the user specifications, the first in-order input determines the other input when the contents in different gaps are linked. The restrictive editor design facilitates the expression of insights and the description of model components. Finally, we designed generation rules to translate the Formal ODD description into NetLogo code. The rules are attached in the Appendix section entitled "Generation Rules of ODD2NetLogo".

After the preliminary language and tool design, we selected suitable social simulation models to act as DSM tests (Step G in Figure 5.1). We chose specific library models to perform the DSM tests. Following the Agile Programming principles [64], we extended the structure and defined additional generation rules when the tool failed to accommodate the test models. We improved the editor in cases it allowed wrong choices or when it was not intuitive. We followed a rule to use as few concepts as possible. The resulting structure (Step E of the Methodology Figure 5.1) is provided in the Documentation folder of the GitHub repository dedicated to the ODD2NetLogo project (https://github.com/uiano/odd2netlogo).

The resulting Formal ODD language extends the ODD protocol by including the experiment design. In the ODD2NetLogo tool, the user specifies both the model purpose and the experiment design. As a result, the combination of these formal ODD specifications exposes potential deviation between the model's intended and actualised role. The experiment design specifications generate the NetLogo Behavioral space, which is used to automate simulation scenarios. The description of the experiment design improves the simulation model transparency. Furthermore, gaining access to the experiment design provides means to understand the extent to which the model was studied. Finally, in our tool, we distinguish between model exploration and experimentation. The model exploration component includes features for manual experiments, while the model experimentation component aims at the design of bulk simulations.

5.5 Feedback for the tool (Steps H and 11)

We used the MARG model and the MARG experience to evaluate the DSM tool (Step H of Figure 5.1). We extended the tool to accommodate the MARG model with the test models. In addition, we used our case study experience in the DSM tool design to accommodate partial solutions for the problems we faced. We have described in Paper E how we accomplished the latter, but a summary is included in the next paragraph.

Our experience with the MARG model highlights the importance of conceptual clarification and the potential risks when the conceptual model terminology is ambiguous. In the literature search, I observed that researchers emphasise the importance of verifying the ABM rules. However, conceptual ambiguity is also prominent in the definition of entities and their properties. Paper E contains an example of this, where the entity student is used in dissimilar contexts in two different ABMs. To improve conceptual clarification and better model context and physics description, the ODD2NetLogo contains two significant features: the formal link of entities with properties and the deconstruction of behavioural rules into elemental actions. The latter would have made an impact when we were creating the bullying model. Such descriptions unavoidably deconstruct the concept of bullying and the MARG model property "bullied" given that the conceptual model is designed in the presence of knowledge holders.

5.6 Solution Summary

The following list summarises the function of our DSM tool:

- The users (knowledge holders and modellers) express the model description in the Formal ODD language using the ODD2NetLogo editor.
- The ODD2NetLogo automatically generates the NetLogo code with a click.
- The Formal ODD description acts as model documentation.

The Formal ODD design, Formal ODD language and our particular implementation enhance model transparency, provide a platform to accommodate participatory approaches, optimise the model creation process, and address the problem statement. The detailed benefits of the solution are listed as follows [1]:

- Modelling choices and discussions are performed only at the conceptual model level (Formal ODD design concept).
- The Formal ODD tool empowers knowledge holders to co-create the simulation model in line with the values of participatory approaches (Formal ODD design concept).
- The input of the knowledge holders is retained in the computer model description in the absence of third-party involvement (Formal ODD design concept).
- Incomplete descriptions do not generate code. Consequently, the model documentation is complete and accurate leading to enhanced model transparency (Formal ODD design concept).
- The domain language is user-friendly and familiar to social simulation researchers (Formal ODD design concept).
- The model creation process is optimised by removing the coding, verification, and model documentation processes (Formal ODD design concept).
- The Formal ODD language extends the ODD protocol in its capacity to accurately describe the model exploration and model experimentation processes (Formal ODD language).
- The Formal ODD language exposes the programmatic formulation and model context information resulting in a more concrete simulation model description (Formal ODD language).
- The ODD2NetLogo implementation offers means of conceptual clarification (ODD2NetLogo implementation).
- In the ODD2NetLogo implementation, updating and extending simulation models do not require model integration (ODD2NetLogo implementation).

[1] The component linked with the benefit is attributed in the parenthesis

- The ODD2NetLogo implementation further reduces the model creation time by automatically generating Formal ODD sections when this is appropriate (see Paper E) (ODD2NetLogo implementation).

5.7 Dissemination and Use (Step I)

We presented a tutorial on the ODD2NetLogo during the Social Simulation week. The tool, a written tutorial and documentation are provided in the GitHub repository dedicated to the project (https://github.com/uiano/odd2netlogo). Please note that we update the tool and the documentation/tutorial at different times. Hence, there may be some misalignment in the documentation and tutorial descriptions.

Chapter 6

Discussion

> But I do like clarity and exact thinking. And I believe that (it is) very important to mankind. Because when you allow yourself to think inexactly, your prejudices, your bias, your self-intuition comes in ways you don't notice. And you do bad things without knowing that you're doing them. Self-deception is very easy. So that I do think clear thinking immensely important.
>
> *Bertrand Russell*
> *Interview shown in [132]*

In this Chapter, I identify alternative routes to approach the stated problem and the limitations of the presented work. Finally, I discuss the issue of precision as it is one of the foundations for the described methodology.

6.1 Precision vs Imprecision

The approach we followed to address the stated problem relies on precision. Paper C includes arguments as to why and how concrete and precise terms will improve our social simulation models. However, I felt the need to extend the discussion to realms outside of the academic world. If I were to describe precision, I would say it reflects an intentional effort to use the least ambiguous terms to convey a message. In addition, a precise description is both sufficiently complete and concrete. Keeping this in mind, formal languages, such as the Formal ODD, are considered more precise than informal ones, such as natural languages or the traditional ODD protocol. In this section, I explore preferences and choices related to precision and imprecision in everyday communication and within decision-making institutions to initiate a bigger discussion.

Precision is crucial in scientific research. We begin our articles and books with concept definitions in an effort to be precise in our content. This serves the peer review needs for transparency and disambiguates our arguments. Some scientific fields such as mathematics, which use formal languages, have higher precision requirements compared to other fields, such as social sciences, which have to contend with the imprecision of a natural language. For instance, this study highlighted that researchers have been using the concept of bullying without fully acknowledging its ambiguity, which is partly due to the imprecise and abstract definitions of the term.

Precision is a vital component of scientific research, but in everyday communication, imprecision is often the norm. During this study, I encountered difficulty in getting friends and family to use precise language, indicating a lack of appreciation for precision in everyday interactions. Upon further investigation, I discovered that imprecise language is often used in communication as nonverbal cues, such as the vocal tone, and facial expressions also play a significant role in conveying meaning [133]. The cited study states that nonverbal cues can account for more than 90% of our communication. This finding highlights the complexity of communication and the importance of understanding that precision is not always necessary or even desirable because of the additional effort it requires in everyday interactions.

While imprecision in communication can be common and not necessarily detrimental, it is sometimes used intentionally for personal, institutional, or market gain. On a personal level, it can be used as a strategy of deception [134]. Larger institutions and markets may incorporate imprecise statements as a strategy to achieve their goals. Some examples include the Federal Reserve's vague announcements in 1989 to achieve inflation goals [135] and the ambiguities in Turkish laws, which allowed the emergence of private markets in Istanbul [136]. As Nassar and Nora note, in Lebanon, legal ambiguities were strategically permitted to avoid drawbacks from the legalisation of Syrian refugees [137]. This allowed Lebanon to maintain neutrality in the war while technically abiding by international customary law regarding refugees [137].

The modelling framework introduced in this dissertation can help us grasp the consequences of imprecise descriptions. An informal definition of bullying in bullying interventions, such as the KIVA interventions, or in the Norwegian law allows the prescription for multiple models of reactions for the same observed behaviours or situations. By making imprecise definitions, stakeholders have more power to interpret where they can apply the bullying interventions. On the other hand, a precise description allows the writer to restrict the scope of a bullying intervention.

Examples of the effects of precision and imprecision can be found in the law literature, where imprecision can have both positive and negative consequences [138]. Based on Gillian K. Hadfield's work, imprecision can correct errors made by lawmakers, address diversity, and offer flexibility for future needs. On the other hand, she recognises that it leads to "arbitary enforcement" and a sense of unfairness and insecurity.

Drawing from the literature and my discussions with different knowledge holders and stakeholders, in the context of antibullying programs, imprecision in bullying definitions has a dual effect. It can protect more students by allowing additional behaviours to be categorised as bullying, and it can lead to arbitrary judgments and accusations of bullying behaviour.

In this research, precision is valued due to its scientific nature. My motivation in this section was to extend the discussion outside academia, especially in fields such as law and decision-making. Ultimately, both precision and imprecision have their own benefits and drawbacks. It is important to find a balance between freedom of interpretation and precision in prescribed reactions to different situations. In the future, it would be beneficial to participate in more transparent discussions on the desired level of precision in various fields, and how it can impact decision-making in everyday life.

6.2 ODD2NetLogo, Knowledge Representation and Knowledge Engineering

Every form of insight expression, whether in a conceptual model, a computer model, or natural language, as seen with original knowledge insights, is a representation of knowledge. However, the term "Knowledge Representation" has a specific meaning in Computer Science as articulated by Sowa in his book [7] and Davis in his renowned article [8]. They both outline conditions that disqualify the original knowledge insights as a "Knowledge Representation". Knowledge engineers, as per Sowa, play a crucial role in transforming original insights into a machine-readable form primed for computations. This transformation requires advanced skills. ODD2NetLogo serves to convert knowledge insights provided by knowledge holders into a computer model description, leading us to ask, "Have we genuinely automated knowledge engineering?" The quick response is no, and this was never our objective.

To elaborate, let's first discuss ODD2NetLogo's treatment of ontology and the application of logic. One advantage of ODD2NetLogo is that it facilitates the specification of formal ontologies. As a language, it already carries some ontological commitments. In the NetLogo language, the "world" consists of environmental entities, regular entities, and networks. Due to its inheritance from NetLogo and the need for generation rules to accommodate NetLogo, ODD2NetLogo also takes on these ontological commitments. Meanwhile, the Agent-Based framework introduces similar ontological commitments, leading to substantial influence and their inheritance in the ODD protocol, a documentation tool for social simulation. Consequently, the Formal ODD, exemplified by the ODD protocol, adopts all these commitments. The implication of this adoption is that users must align their knowledge insights with ODD2NetLogo's ontological commitments, potentially disrupting the original knowledge insights' ontology. However, this disruption is unavoidable within the modelling team's work.

In terms of logic, ODD2NetLogo employs a combination of procedural logic found in programming languages and mathematical logic, allowing mathematical function formulations. Simulations enabled by the tool's output, which is the computer model description, can then produce new knowledge based on our knowledge insights, formalised in the initial ODD2NetLogo input. The place of ontology, the use of logic, and the fact that ODD2NetLogo generates simulation models that act to generate new knowledge are the three elements Sowa considers important for the classification of ODD2NetLogo as a Knowledge Representation language [7].

By offering a structured platform for formal ontologies with its inherent ontological commitments, ODD2NetLogo assumes some responsibilities typical of the knowledge engineer role but does not automate the process of knowledge engineering. The weight of managing their own knowledge still falls upon the knowledge holders. We designed ODD2NetLogo, not as a mechanism for automating knowledge engineering, but as a platform for knowledge holders to voice their insights. By addressing the technical prerequisites, ODD2NetLogo empowers knowledge holders to create a readable form of their insights written in an executable knowledge representation format. As a whole, employing ODD2NetLogo in simulation model creation diminishes the assumptions typically made by knowledge engineers when transforming informal specifications into formal ones. However, ODD2NetLogo does not question insights expressed in the required format. Overall, ODD2NetLogo would be a great tool in combination with the work of a knowledge engineer but can provide results as it is.

We did envision knowledge holders to leverage this platform to identify the deficiencies in their knowledge and question their insights' validity. This self-examination arises when knowledge holders input their insights into ODD2NetLogo and strictly adhere to their ontology, potentially revealing inconsistencies in practice. The high level of precision required for ODD2NetLogo input additionally contributes to this effort. As a consequence, ODD2NetLogo evolves into a platform for discussion that can foster agreement on issues such as the ontology of bullying. However, it is not possible to predict the extent to which ODD2NetLogo will unveil tacit and deficient knowledge.

6.3 Alternatives

I have divided the section into three parts to explore methodological alternatives for the misalignment of knowledge-holder insights with their implementations. In the first part, I refer to the general design methodology. In the second part, I discuss alternatives to the design choices of the DSM tool. The final part consists of alternatives to the bullying model development.

- **Methodological Alternatives** The design methodology requires an evaluation component. Early in the project, we approached evaluation using a case study. We decided to create a new social simulation model to experience the modelling process from start to finish. However, an alternative way is to use an existing social simulation model for the evaluation. Following this alternative, time invested in the development of the social simulation model would have been used to improve the tool's functionality.

  Another alternative would have been to spend more time exploring the social conflict literature. Articles such as [139]) would have pointed to the problem of concept ambiguity earlier in the project.

  One more idea would have been to focus on the case study and later on the tool design. This would have provided more time to reflect on possible approaches to address the problem and not rely on a tentative problem and solution. From a project management perspective, it would have made the research less complex.

A different option would have been to focus on completing the case study simulation model before moving to the DSM tool, which could have resulted in a fuller action-oriented approach to the modelling. Upon discovering the ambiguity issue, we could have attempted to resolve it with the community and probably would have ended up with a model of concrete actions as we suggested in Paper C.

A final alternative would have been to develop the proof of concept for the tool before the start of the model creation process and use it to support the process. The development of a functional tool would have taken a lot of time. By the time the tool would be ready, the COVID-19 circumstances would have prevented a functional participatory approach. Notwithstanding the actual circumstances, we would have less time for model development, preventing the reflection process that exposed the bullying ambiguity. As a result, I expect that we would have ended up with a bullying model, which from what I explained in the thesis, is not a good idea.

- **Design Alternatives** One of the most interesting explorations of alternative steps is the design alternatives for our DSM tool. Using a general-purpose language would probably offer more opportunities to integrate models but also require more time to build the same functionality, as NetLogo is very accommodating to coding ABMs.

  Regarding the domain language, the reaction to alternative paths is more complicated. An idea would be to build our domain language to describe ABMs or specifically target social conflict subjects conceptually. Building such as language from scratch would have been time-consuming, required multiple tests to check its eligibility, and would have increased its learning curve. On the other hand, it would have been suitable for the creation of conflict resolution models.

  One more option would have been to use existing frameworks for conceptual descriptions or meta-models. In the literature search for Paper A, we found out that the most suitable meta-models did not provide automated code generation, so the first step in this alternative would have been to create the generation rules. However, many meta-models provide semi-automatic code generation, which implies that we would need less time to build the transformation rules. A prominent candidate would have been the meta-model MAIA [140]. In addition to being a framework to assist the conceptualisation of models around social institutions, it helps to structure the data collection so that qualitative data can be directly applied to the conceptual model description [108]. To use the MAIA meta-model, we would perform additional analysis to check its suitability for models of social conflict and then add transformation rules to adjust its functionality to the DSM tool.

- **Modelling Alternatives** The case study of bullying offered insights into unnoticed modelling issues such as conceptual ambiguities. Still, it did not provide a good modelling subject to experience a complete model creation process. An alternative case study would have provided this opportunity should other issues not arise. This probably would have not contributed to the theoretical exploration presented in this thesis, but it could have offered additional interactions to test the tool in action.

A modelling alternative to the development of the bullying model would be to use existing conceptual frameworks with the knowledge holders. I believe this would have demystified the situation sooner.

Finally, using the Olweus definition (see [74]) instead of the Schott and Søndergaard, given its popularity, would have left the issue of ambiguity unnoticed, leading to a bullying model, which, as mentioned in the Methodological alternatives, would not be a good idea.

6.4 Limitations and Future Steps

The current version of the DSM tool is in the state of a proof of concept. I distinguish between inherent and version limitations, followed by a plan for future steps.

Most of the inherent limitations of the DSM tool relate to design choices. By definition, the DSM tool inherits the same limitations as NetLogo in its applicability, efficiency and expressivity as a programming language. One important thing to consider regarding our tool is that the defined transformations from the conceptual description to the model code may limit the code diversity and, for some models, may lack optimisation in algorithmic complexity. At the same time, we have not designed transformations from the computer code to the conceptual model description, which means that it is not possible to generate the description for existing models.

I identify two ways ODD is constrained as a domain language towards conceptual descriptions. First, whereas it can accommodate pluralism in the modelled concepts, it cannot offer solutions to conceptual disagreements. In other words, agreement over the conceptual model should precede the description in the DSM tool. Second, ODD is primarily a documentation protocol. It is not meant to structure qualitative data. As a result, we cannot predict how intuitive it is to use as a conceptual framework from knowledge holders.

In the thesis, I define knowledge holders as anyone with knowledge in a domain. This definition includes children marginalised in their school environment. Nevertheless, it would be almost impossible for an average child to use the ODD to describe their insights. In those cases, traditional data collection methods work better. It would be more accurate to say that ODD2NetLogo is geared towards knowledge holders with accumulated knowledge and does not handle "data acquisition".

A final limitation is that the Formal ODD description can be lengthy and the writing process tedious. However, traditional coding and documentation processes take more time.

The limitations of the DSM tool's current state, including provisional steps, are the following:

- More work is needed to identify all possible elemental actions and create specifications to limit the user input when describing behaviours.
- The user interface is not thoroughly validated. We have conducted two rounds of interface assessment by modellers. The most reasonable next step would be to plan workshops with diverse audiences for further evaluation. One aspect that needs to be assessed is the user interface's performance in the description of qualitative data.

- We have not assessed the learning curve for the DSM tool. However, we estimate that modellers can use it without any issues, especially given the provided tutorials and documentation. Future versions should be more readily usable for broader audiences.
- Regarding the ODD2NetLogo documentation, the current files do not describe the formal semantics we used. However, it is possible to access the transformation rules in the downloadable software.

One final limitation towards the overall approach is that we do not discuss which insights should be implemented in the model. I realise that academic knowledge-holder insights may be more refined compared to non-academic knowledge-holder insights. Still, I would argue that the modelling team should not necessarily be responsible for the modelled insights, especially in the context of participatory approaches. As Sam states in [121], we should consider non-academic knowledge holders as people with "exclusive access to their social reality", making their knowledge more reliable.

Chapter 7

Summary

This dissertation approaches the issue of misalignment between knowledge holder insights and their implementation in the simulation models. Figure 7.1 provides an overview of the accumulated knowledge and contributions.

Our experience with the case study highlighted the importance of choosing clear concepts and placed the starting point of the modelling process in the conception of the insights from the knowledge holder (Stage of Bullying Experience and Mental Model Stage in Figure 7.1). The bullying research showcases that the same student interaction can be perceived, conceptualised and communicated dissimilarly by different people. The linguistic concepts we use can either blur or clarify our descriptions. For instance, the use of the term "bullying" increases the chances of misinterpretation by the modelling team. In the context of social simulation, this means the modelling team can misrepresent the knowledge holder insights in the conceptual model description (Transition 0). The transition from the conceptual to the computer model description (Transition 1) introduces assumptions and coding errors that can lead to a further deviation of the model content from the initial insights. The resulting simulation model, which we access through our simulations and data analysis, might differ significantly from the mental models of the knowledge holders. To address this issue, the modelling team typically conducts the Knowledge verification process. During this process, knowledge holders check whether their insights are implemented correctly. However, since most knowledge holders do not have programming skills, they assess model descriptions on the conceptual model level, such as the ODD model documentation, and not in the code. Errors in the documentation process can make the actual implementation more inaccessible.

We used a DSM approach to address the problem with the development of a Formal ODD concept, a Formal ODD language and the ODD2NetLogo tool that incorporates the Formal ODD language. The tool receives model descriptions in the Formal ODD language and automates Transition 1 by generating the computer model description. Thus, the Formal ODD description acts both as a conceptual model prescription and model documentation and allows knowledge holders to read, describe and edit their knowledge insights in a relatively intuitive language. Additional benefits include the reduction of the model creation time and a platform for conceptual clarification.

Finally, we created a new framework to describe the modelling process by emphasising the difference between models (M0) and model descriptions (M1), and introduced the concepts of model physics and model context. Figure 7.1 incorporates the OMG hierarchy levels, which we used in our framework. It becomes clear that we transition from the conceptual to the computer model using descriptions (M1). The different languages (M2) change our capacity to describe the model context and model physics.

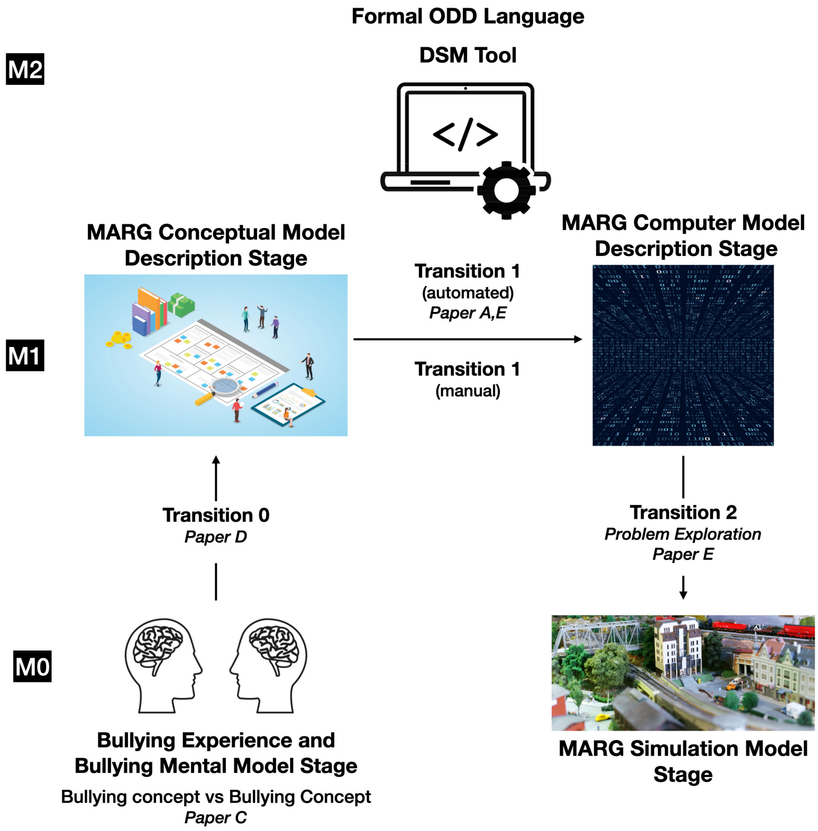


Figure 7.1: Summary of the presented work

With the implementation of the planned steps, the ODD2NetLogo tool will fulfil the benefits of the Formal ODD design. Thus, the explored problem will be partially solved, using the word "partial" to account for the described limitations. Although our approach is geared towards action-oriented research, the work shows that sometimes we must step back and reflect. Policymakers require confidence from scientists, and many people consider science a slow process. Despite the high demands and criticism, my firm belief is that responsibility should be our first priority. It is my deepest hope that the theoretical frameworks developed in this work will move the modelling discipline a step further towards quality modelling. Without quality, our proposals such as in the case of bullying interventions may end up in frustrated stakeholders and unresolved issues. After all, quality is the promise of science to policymakers and society at large.

Bibliography

[1] Steven Kelly and Juha-Pekka Tolvanen. *Domain-specific modeling : enabling full code generation*. Wiley, Hoboken, N.J, 2008.

[2] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3):45–77, 2007. doi: 10.2753/MIS0742-1222240302.

[3] Kathy Kotiadis and Stewart Robinson. Conceptual modelling: Knowledge acquisition and model abstraction. In *2008 Winter Simulation Conference*, pages 951–958, 2008.

[4] Catherine Tinsley. Models of conflict resolution in Japanese, German, and American cultures. *Journal of Applied Psychology*, 83:316–323, 1998.

[5] Carlos Miguel Lemos, Hélder Coelho, and Rui J. Lopes. Agent-based Modeling of Social Conflict, Civil Violence and Revolution: State-of-the-art-review and Further Prospects. In *European Workshop on Multi-Agent Systems*, 2013.

[6] Volker Grimm, Jacqueline Augusiak, Andreas Focks, Béatrice M. Frank, Faten Gabsi, Alice S. A. Johnston, Chun Liu, Benjamin T. Martin, Mattia Meli, Viktoriia Radchuk, Pernille Thorbek, and Steven F. Railsback. Towards better modelling and decision support: Documenting model development, testing, and analysis using TRACE. *Ecological Modelling*, 280:129–139, 2014.

[7] John F. Sowa. *Knowledge representation : logical, philosophical, and computational foundations*. Brooks/Cole, Pacific Grove, Calif, 2000. Bibliografi: s. 549-573.

[8] Randall Davis, Howard E. Shrobe, and Peter Szolovits. What is a knowledge representation? *AI Mag.*, 14:17–33, 1993.

[9] Sergei Nirenburg and Victor Raskin. Ontological semantics, formal ontology, and ambiguity. In *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, page 151–161, New York, NY, USA, 2001. Association for Computing Machinery.

[10] Dieter Wolke, Neil Tippett, and Slava Dantchev. Bullying in the family: sibling bullying. *The Lancet Psychiatry*, 2(10):917–929, 2015.

[11] Shelley Hymel and Susan M Swearer. Four decades of research on school bullying: An introduction. *American Psychologist*, 70(4):293, 2015.

[12] II James E. Bartlett and Michelle E. Bartlett. Workplace Bullying: An Integrative Literature Review. *Advances in Developing Human Resources*, 13(1):69–84, 2011.

[13] Robert Slonje, Peter K. Smith, and Ann Frisén. The nature of cyberbullying, and strategies for prevention. *Computers in Human Behavior*, 29(1):26–32, 2013. Including Special Section Youth, Internet, and Wellbeing.

[14] Gordon W. Blood and Ingrid M. Blood. Long-term consequences of childhood bullying in adults who stutter: social anxiety, fear of negative evaluation, self-esteem, and satisfaction with life. *Journal of fluency disorders*, 50:72–84, 2016.

[15] Dieter Wolke and Suzet Tanya Lereya. Long-term effects of bullying. *Archives of disease in childhood*, 100(9):879–885, 2015.

[16] Dewey Cornell, Anne Gregory, Francis Huang, and Xitao Fan. Perceived prevalence of teasing and bullying predicts high school dropout rates. *Journal of Educational Psychology*, 105(1):138, 2013.

[17] Gina Vega and Debra R. Comer. Sticks and stones may break your bones, but words can break your spirit: Bullying in the workplace. *Journal of business ethics*, 58(1):101–109, 2005.

[18] Ingrid Lund. Mobbing i høyere utdanning, fleip eller fakta. Report ISBN 978-82-7117-861-1, PULS, University of Agder, 2017.

[19] Lov om universiteter og høyskoler (universitets- og høyskoleloven). https://lovdata.no/NL/lov/2005-04-01-15/§4-3, 2005. Legal Rule or Regulation.

[20] UHRs arbeidsgruppe mot mobbing og trakassering. http://www.uhr.no/aktuelt_fra_uhr/uhrs_arbeidsgruppe_mot_mobbing_og_trakassering, March 2018. Press Release.

[21] Michaela Bruckmayer and Silvia Galimberti. Anti-bullying Practices from the Repository of the European Platform for Investing in Children (EPIC) . Report, European Commission, 2020.

[22] European Commission. An official website of the European Union. https://ec.europa.eu/info/index_en. Last accessed: 08.12.2021.

[23] Rachel C. Vreeman and Aaron E. Carroll. A systematic review of school-based interventions to prevent bullying. *Arch Pediatr Adolesc Med*, 161(1):78–88, 2007. Vreeman, Rachel C Carroll, Aaron E Journal Article Review Systematic Review United States Arch Pediatr Adolesc Med. 2007 Jan;161(1):78-88. doi: 10.1001/archpedi.161.1.78.

[24] Peter K. Smith, Katerina Ananiadou, and Helen Cowie. Interventions to reduce school bullying. *The Canadian Journal of Psychiatry*, 48(9):591–599, 2003.

[25] Seokjin Jeong and Byung Hyun Lee. A Multilevel Examination of Peer Victimization and Bullying Preventions in Schools. *Journal of Criminology*, 2013:10, 2013.

[26] Shelley Hymel and Susan M. Swearer. Four decades of research on school bullying: An introduction. *American Psychologist*, 70:293–299, 2015.

[27] Milda Astrauskaite, Roy M Kern, and Guy Notelaers. An individual psychology approach to underlying factors of workplace bullying. *The Journal of individual psychology*, 70(3):220–244, 2014.

[28] Jon Sutton, Peter K Smith, and John Swettenham. Bullying and 'theory of mind': A critique of the 'social skills deficit'view of anti-social behaviour. *Social development*, 8(1):117–127, 1999.

[29] Liang Huang. Exploring the relationship between school bullying and academic performance: the mediating role of students' sense of belonging at school. *Educational Studies*, 48(2):216–232, 2022. doi: 10.1080/03055698.2020.1749032.

[30] Anol Bhattacherjee. Social Science Research: Principles, Methods, and Practices. https://digitalcommons.usf.edu/oa_textbooks/3/, 2012. Last Accessed: 15/09/2022.

[31] José Manuel Galán, Luis R. Izquierdo, Segismundo S. Izquierdo, José Ignacio Santos, Ricardo del Olmo, Adolfo López-Paredes, and Bruce Edmonds. Errors and Artefacts in Agent-Based Modelling. *Journal of Artificial Societies and Social Simulation*, 12(1):1, 2009.

[32] Gerrit Van der Waldt. Constructing conceptual frameworks in social science research. *The Journal for Transdisciplinary Research in Southern Africa*, 16(1), 2020.

[33] Nurit Shnabel, Arie Nadler, Daphna Canetti-Nisim, and Johannes Ullrich. The Role of Acceptance and Empowerment in Promoting Reconciliation from the Perspective of the Needs-Based Model. *Social Issues and Policy Review*, 2(1):159–186, 2008. https://doi.org/10.1111/j.1751-2409.2008.00014.x.

[34] Max Weisbuch and Kristin Pauker. The Nonverbal Transmission of Intergroup Bias: A Model of Bias Contagion with Implications for Social Policy. *Social Issues and Policy Review*, 5(1):257–291, 2011. https://doi.org/10.1111/j.1751-2409.2011.01032.x.

[35] Jay J. Van Bavel, Elizabeth A. Harris, Philip Pärnamets, Steve Rathje, Kimberly C. Doell, and Joshua A. Tucker. Political Psychology in the Digital (mis)Information age: A Model of News Belief and Sharing. *Social Issues and Policy Review*, 15(1):84–113, 2021. https://doi.org/10.1111/sipr.12077.

[36] William M. Snyder and Thomas G. Cummings. Organization Learning Disorders: Conceptual Model and Intervention Hypotheses. *Human Relations*, 51(7):873–895, 1998. doi: 10.1177/001872679805100702.

[37] Piotr Tarka. An overview of structural equation modeling: its beginnings, historical development, usefulness and controversies in the social sciences. *Quality & Quantity*, 52(1):313–354, 2018.

[38] Charles A. Lave and James G. March. *An introduction to models in the social sciences*. Harper & Row, New York, 1975.

[39] Jay W. Forrester. System dynamics—a personal view of the first fifty years. *System Dynamics Review*, 23(2-3):345–358, 2007. https://doi.org/10.1002/sdr.382.

[40] A. Edelmann, T. Wolff, D. Montagne, and C. A. Bail. Computational Social Science and Sociology. *Annu Rev Sociol*, 46(1):61–81, 2020.

[41] Marco Buzzoni. Thought Experiments and Computer Simulations. In Lorenzo Magnani and Claudia Casadio, editors, *Model-Based Reasoning in Science and Technology*, pages 57–78. Springer International Publishing, 2016.

[42] Johannes Lenhard. Thought Experiments and Simulation Experiments: Exploring Hypothetical Worlds. In Michael T Stuart, Yiftach Fehige, and James Robert Brown, editors, *The Routledge companion to thought experiments*. Routledge London, 2018.

[43] Keith Dowding and Enzo Lima. Models and Stories in Social Science. 10.13140/RG.2.2.13480.26886, 2018. Unpublished Work.

[44] Bruce Edmonds, Christophe Le Page, Mike Bithell, Edmund Chattoe-Brown, Volker Grimm, Ruth Meyer, Monta, ntilde, Cristina ola Sales, Paul Ormerod, Hilton Root, and Flaminio Squazzoni. Different Modelling Purposes. *Journal of Artificial Societies and Social Simulation*, 22(3):6, 2019.

[45] Joshua M. Epstein. Why Model? *Journal of Artificial Societies and Social Simulation*, 11(4):12, 2008.

[46] Joshua M. Epstein. *Generative Social Science: Studies in Agent-Based Computational Modeling*, volume 21 of *Princeton Studies in Complexity ;*. Princeton University Press, Princeton, NJ, course book. edition, 2012.

[47] Corinna Elsenbroich. Social Simulation and Online Research Methods. In Nigel G. Fielding, Raymond M. Lee, and Grant Blank, editors, *The SAGE Handbook of Online Research Methods*. SAGE Publications Ltd, 2017.

[48] Andrei Borshchev and Alexei Filippov. From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools. In *Proceedings of the 22nd international conference of the system dynamics society*, volume 22. Citeseer, 2004.

[49] Gabriel Wurzer, Kerstin Kowarik, and Hans Reschreiter. *Agent-based modeling and simulation in archaeology*. Springer, 2015.

[50] Jule Schulze, Birgit Müller, Jürgen Groeneveld, and Volker Grimm. Agent-based modelling of social-ecological systems: achievements, challenges, and a way forward. *Journal of Artificial Societies and Social Simulation*, 20(2), 2017.

[51] Parsa Pouladi, Abbas Afshar, Mohammad Hadi Afshar, Amir Molajou, and Hamid Farahmand. Agent-based socio-hydrological modeling for restoration of Urmia Lake: Application of theory of planned behavior. *Journal of Hydrology*, 576:736–748, 2019.

[52] Elizabeth Hunter, Brian Mac Namee, and John D Kelleher. Using a socioeconomic segregation burn-in model to initialise an agent-based model for infectious diseases. *Journal of Artificial Societies and Social Simulation*, 21(4), 2018.

[53] Petra Ahrweiler and Nigel Gilbert. The Quality of Social Simulation: An Example from Research Policy Modelling. In Marijn Janssen, Maria A. Wimmer, and Ameneh Deljoo, editors, *Policy Practice and Digital Science: Integrating Complex Systems, Social Simulation and Public Administration in Policy Research*, pages 35–55, Cham, 2015. Springer International Publishing.

[54] Claudius Gräbner. How to relate models to reality? An epistemological framework for the validation and verification of computational models. *Journal of Artificial Societies and Social Simulation*, 21(3), 2018.

[55] Olivier Barreteau, Pieter Bots, Katherine Daniell, Michel Etienne, Pascal Perez, Cécile Barnaud, Didier Bazile, Nicolas Becu, Jean-Christophe Castella, William's Daré, and Guy Trebuil. Participatory Approaches. In Bruce Edmonds and Ruth Meyer, editors, *Simulating Social Complexity: A Handbook*, pages 197–234, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[56] S. Thoyer, Sylvie Morardet, and P. Rio. A short answer to Bruce Edmonds's commentary on Sophie Thoyer, Sylvie Morardet, Patrick Rio, Leo Simon, Rachel Goodhue and Gordon Rausser (2001)" A Bargaining model to simulate negotiations between water users". *Journal of Artificial Societies and Social Simulation*, 4(2):2, 2001.

[57] Igor Nikolic, Koen H. van Dam, and Julia Kasmire. Practice. In Koen H. van Dam, Igor Nikolic, and Zofia Lukszo, editors, *Agent-Based Modelling of Socio-Technical Systems*, pages 73–137, Dordrecht, 2013. Springer Netherlands.

[58] Nuno David, Nuno Fachada, and Agostinho C. Rosa. Verifying and Validating Simulations. In Bruce Edmonds and Ruth Meyer, editors, *Simulating Social Complexity: A Handbook*, pages 173–204, Cham, 2017. Springer International Publishing.

[59] Sukaina Bharwani. *Adaptive knowledge dynamics and emergent artificial societies: ethnographically based multi-agent simulations of behavioural adaptation in agro-climatic systems*. PhD thesis, University of Kent, 2004.

[60] Nilesh Anand, David Meijer, J. H. R. van Duin, Lóránt Tavasszy, and Sebastiaan Meijer. Validation of an agent based model using a participatory simulation gaming approach: The case of city logistics. *Transportation Research Part C: Emerging Technologies*, 71:489–499, 2016.

[61] Romain Reuillon, Clara Schmitt, Ricardo De Aldama, and Jean-Baptiste Mouret. A new method to evaluate simulation models: The calibration profile (cp) algorithm. *Journal of Artificial Societies and Social Simulation*, 18(1):12, 2015.

[62] Fran Baum, Colin MacDougall, and Danielle Smith. Participatory action research. *Journal of epidemiology and community health*, 60(10):854, 2006.

[63] Tony Clark, Paul Sammut, and James Willans. Applied Metamodelling: A Foundation for Language Driven Development (Third Edition). https://arxiv.org/abs/1505.00149, 2015.

[64] Daniel Moyo, Abdallah K. Ally, Alan Brennan, Paul Norman, Robin C. Purshouse, and Mark Strong. Agile Development of an Attitude-Behaviour Driven Simulation of Alcohol Consumption Dynamics. *Journal of Artificial Societies and Social Simulation*, 18(3):10, 2015.

[65] Act on working environment, working hours and job protection etc. (Working Act). https://lovdata.no/dokument/NLE/lov/2005-06-17-62, 2005 (Last update on 2019).

[66] Speak Up. https://www.uia.no/en/about-uia/speak-up. Last accessed: 27/08/2019.

[67] Who can report? https://www.uia.no/en/about-uia/speak-up/who-can-report. Last accessed: 27/08/2019.

[68] Handling of reports received. https://www.uia.no/en/about-uia/speak-up/hvordan-varsle/handling-of-reports-received. Last accessed: 27/08/2019.

[69] Hvordan varsle. https://www.uia.no/en/about-uia/speak-up/hvordan-varsle. Last accessed: 27/08/2019.

[70] Are there risks involved in reporting? https://www.uia.no/en/about-uia/speak-up/hvordan-varsle/are-there-risks-involved-in-reporting 1/2. Last accessed: 27/08/2019.

[71] False accusation. https://www.uia.no/en/about-uia/speak-up/hvordan-varsle/false-accusation. Last accessed: 27/08/2019.

[72] Guidelines for case handling when students feel bullied/harassed. https://www.uia.no/en/student/retningslinjer-for-saksbehandling-naar-studenter-mener-seg-mobbet-trakassert. Last accessed: 27/08/2019.

[73] Robin M. Schott and Dorte M. Søndergaard. *School Bullying: New Theories in Context*. Cambridge University Press, Cambridge, 2014.

[74] Dan Olweus. Bullying at School. In L. Rowell Huesmann, editor, *Aggressive Behavior: Current Perspectives*, pages 97–130, Boston, MA, 1994. Springer US.

[75] Christina Salmivalli, Kirsti Lagerspetz, Kaj Björkqvist, Karin Österman, and Ari Kaukiainen. Bullying as a Group Process: Participant Roles and their Relations to Social Status within the Group. *Aggressive Behavior*, vol. 22(1):pp. 1–15, 1996.

[76] Themis D. Xanthopoulou, Ivan Puga-Gonzalez, F. LeRon Shults, and Andreas Prinz. Modeling Marginalization: Emergence, Social Physics, and Social Ethics of Bullying. In *2020 Spring Simulation Conference (SpringSim)*, pages 1–12, 2020.

[77] Yoshinobu Maeda, Kazuya Anezaki, and Yumi Takeuchi. An agentbased model for simulating the group dynamics involved in excluding a minority. In *Proc. 1st World Congress on Social Simulation*, volume 1, pages 79–86, 2006.

[78] John W. Thibaut and Harold H. Kelley. *The social psychology of groups.* John Wiley & Sons, INC., New York, 1959.

[79] Themis D. Xanthopoulou, Andreas Prinz, and F. LeRon Shults. The Problem with Bullying: Lessons Learned from Modelling Marginalization with Diverse Stakeholders. In Marcin Czupryna and Bogumił Kamiński, editors, *Advances in Social Simulation*, pages 289–300. Springer International Publishing, 2022.

[80] Faye Mishna, Iolanda Scarcello, Debra Pepler, and Judith Wiener. Teachers' Understanding of Bullying. *Canadian Journal of Education / Revue canadienne de l'éducation*, 28(4):718–738, 2005.

[81] Tracy Vaillancourt, Patricia McDougall, Shelley Hymel, Amanda Krygsman, Jessie Miller, Kelley Stiver, and Clinton Davis. Bullying: Are researchers and children/youth talking about the same thing? *International Journal of Behavioral Development*, 32(6):486–495, 2008.

[82] Jeffrey W. Cohen and Robert A. Brooks. The Construction of School Bullying as a Social Problem. In A. Javier Treviño, editor, *The Cambridge Handbook of Social Problems*, volume 2, pages pp. 237–252, Cambridge, 2018. Cambridge University Press.

[83] Christina Unger and Philipp Cimiano. Representing and resolving ambiguities in ontology-based question answering. In *TextInfer@EMNLP*, 2011.

[84] Ladjel Bellatreche, Nguyen Xuan Dung, Guy Pierra, and Dehainsala Hondjack. Contribution of ontology-based data modeling to automatic integration of electronic catalogues within engineering databases. *Computers in Industry*, 57(8):711–724, 2006.

[85] Luis R. Izquierdo, Doina Olaru, Segismundo S. Izquierdo, Sharon Purchase, and Geoffrey N. Soutar. Fuzzy logic for social simulation using netlogo. *Journal of Artificial Societies and Social Simulation*, 18(4):1, 2015.

[86] Tony Lawson. *The nature of social reality: Issues in social ontology*. Routledge, 2019.

[87] Christina Salmivalli, Antti Kärnä, and Elisa Poskiparta. Counteracting bullying in Finland: The KiVa program and its effects on different forms of being bullied. *International Journal of Behavioral Development*, 35(5):405–411, 2011.

[88] KiVa Antibullying Program Parents' guide. http://data.kivaprogram.net/parents/. Last Accessed: June 2020.

[89] Antti Kärnä, Rinus Voeten, Todd Little, Erkki Alanen, Elisa Poskiparta, and Christina Salmivalli. Effectiveness of the KiVa Antibullying Program: Grades 1-3 and 7-9. *Advance online publication*, 00:1–17, 2012.

[90] Miia Sainio, René Veenstra, Gijs Huitsing, and Christina Salmivalli. Same- and Other-Sex Victimization: Are the Risk Factors Similar? *Aggressive behavior*, 38:442–55, 2012.

[91] Annarilla Ahtola, Anne Haataja, Antti Kärnä, Elisa Poskiparta, and Christina Salmivalli. For children only? Effects of the KiVa antibullying program on teachers. *Teaching and Teacher Education*, 28:851–859, 2012.

[92] Dan Olweus. The Olweus Bully/Victim Questionnaire. *British Journal of Educational PsychologyJournal of Adolescent Health*, 1996.

[93] Lisa Feldman Barrett. *How emotions are made : the secret life of the brain*. Houghton Mifflin Harcourt, 2017.

[94] Janice Langan-Fox, Sharon Code, and Kim Langfield-Smith. Team Mental Models: Techniques, Methods, and Analytic Approaches. *Hum Factors*, 42(2):242–271, 2000.

[95] Sondoss Elsawah, Joseph H. A. Guillaume, Tatiana Filatova, Josefine Rook, and Anthony J. Jakeman. A methodology for eliciting, representing, and analysing stakeholder knowledge for decision making on complex socio-ecological systems: From cognitive maps to agent-based models. *Journal of Environmental Management*, 151:500–516, 2015.

[96] Andreas Prinz, Themis Dimitra Xanthopoulou, Terje Gjøsæter, and Birger Møller-Pedersen. On Abstraction in the OMG Hierarchy: Systems, Models, and Descriptions. In *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, MODELS '22, page 322–330, New York, NY, USA, 2022. Association for Computing Machinery.

[97] Teun Schrieks, W. J. Wouter Botzen, Marthe Wens, Toon Haer, and Jeroen C. J. H. Aerts. Integrating Behavioral Theories in Agent-Based Models for Agricultural Drought Risk Assessments. *Frontiers in Water*, 3, 2021.

[98] Hannah Muelder and Tatiana Filatova. One Theory - Many Formalizations: Testing Different Code Implementations of the Theory of Planned Behaviour in Energy Agent-Based Models. *Journal of Artificial Societies and Social Simulation*, 21(4):5, 2018.

[99] Nicholas M. Gotts, George A. K. van Voorn, J. Gareth Polhill, Eline de Jong, Bruce Edmonds, Gert Jan Hofstede, and Ruth Meyer. Agent-based modelling of socio-ecological systems: Models, projects and ontologies. *Ecological Complexity*, 40:100728, 2019.

[100] Annie Waldherr and Nanda Wijermans. Communicating Social Simulation Models to Sceptical Minds. *Journal of Artificial Societies and Social Simulation*, 16(4):13, 2013.

[101] Loet Leydesdorff, Ismael Ràfols, and Staša Milojević. Bridging the divide between qualitative and quantitative science studies. *Quantitative Science Studies*, 1(3):918–926, 2020.

[102] Bruce Edmonds. Using Qualitative Evidence to Inform the Specification of Agent-Based Models. *Journal of Artificial Societies and Social Simulation*, 18(1):18, 2015.

[103] Paola Tubaro and Antonio A. Casilli. "An Ethnographic Seduction": How Qualitative Research and Agent-based Models can Benefit Each Other. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 106(1):59–74, 2010. doi: 10.1177/0759106309360111.

[104] Jesus Rosales-Carreon and Cesar Garcia-Diaz. Exploring Transitions Towards Sustainable Construction: The Case of Near-Zero Energy Buildings in the Netherlands. *Journal of Artificial Societies and Social Simulation*, 18(1):10, 2015.

[105] Keith Goffin and Ursula Koners. Tacit Knowledge, Lessons Learnt, and New Product Development. *Journal of Product Innovation Management*, 28(2):300–318, 2011. https://doi.org/10.1111/j.1540-5885.2010.00798.x.

[106] L. U. Yang and Nigel Gilbert. Getting away from numbers: Using qualitative observation for agent-based modeling. *Advances in Complex Systems*, 11(02):175–185, 2008. doi: 10.1142/S0219525908001556.

[107] J. Gareth Polhill, Lee-Ann Sutherland, and Nicholas Mark Gotts. Using Qualitative Evidence to Enhance an Agent-Based Modelling System for Studying Land Use Change. *Journal of Artificial Societies and Social Simulation*, 13, 2010.

[108] Amineh Ghorbani, Gerard Dijkema, and Noortje Schrauwen. Structuring Qualitative Data for Agent-Based Modelling. *Journal of Artificial Societies and Social Simulation*, 18(1):2, 2015.

[109] Sukaina Bharwani, Monica Coll Besa, Richard Taylor, Michael Fischer, Tahia Devisscher, and Chrislain Kenfack. Identifying Salient Drivers of Livelihood Decision-Making in the Forest Communities of Cameroon: Adding Value to Social Simulation Models. *Journal of Artificial Societies and Social Simulation*, 18(1):3, 2015.

[110] Bruce Edmonds. A Context- and Scope-Sensitive Analysis of Narrative Data to Aid the Specification of Agent Behaviour. *Journal of Artificial Societies and Social Simulation*, 18(1):17, 2015.

[111] Anthony O'Hagan. Expert Knowledge Elicitation: Subjective but Scientific. *The American Statistician*, 73(sup1):69–81, 2019. doi: 10.1080/00031305.2018.1518265.

[112] Maja Schlüter, Andres Baeza, Gunnar Dressler, Karin Frank, Jürgen Groeneveld, Wander Jager, Marco A. Janssen, Ryan R. J. McAllister, Birgit Müller, Kirill Orach, Nina Schwarz, and Nanda Wijermans. A framework for mapping and comparing behavioural theories in models of social-ecological systems. *Ecological Economics*, 131:21–35, 2017.

[113] Timo Szczepanska, Patrycja Antosz, Jan Ole Berndt, Melania Borit, Edmund Chattoe-Brown, Sara Mehryar, Ruth Meyer, Stephan Onggo, and Harko Verhagen. GAM on! Six ways to explore social complexity by combining games and agent-based models. *International Journal of Social Research Methodology*, 25(4):541–555, 2022. doi: 10.1080/13645579.2022.2050119.

[114] CoMSES.Net/OpenABM. https://www.comses.net/. Last Accessed: 20/09/2022.

[115] Joaquim Carvalho et al. Using AgentSheets to teach simulation to undergraduate students. *Journal of Artificial Societies and Social Simulation*, 3(3):2, 2000.

[116] Melissa V. Eitzel, Jon Solera, K. B. Wilson, Kleber Neves, Aaron C. Fisher, André Veski, Oluwasola E. Omoju, Abraham Mawere Ndlovu, and Emmanuel Mhike Hove. Using mixed methods to construct and analyze a participatory agent-based model of a complex Zimbabwean agro-pastoral system. *PLOS ONE*, 15(8):e0237638, 2020.

[117] J. Gary Polhill and Nicholas M. Gotts. Ontologies for transparent integrated human-natural system modelling. *Landscape Ecology*, 24(9):1255, 2009.

[118] Jule Schulze, Birgit Müller, Jürgen Groeneveld, and Volker Grimm. Agent-based modelling of social-ecological systems: achievements, challenges, and a way forward. *Journal of Artificial Societies and Social Simulation*, 20(2), 2017.

[119] Birgit Müller, Friedrich Bohn, Gunnar Dreßler, Jürgen Groeneveld, Christian Klassert, Romina Martin, Maja Schlüter, Jule Schulze, Hanna Weise, and Nina Schwarz. Describing human decisions in agent-based models – ODD + D, an extension of the ODD protocol. *Environmental Modelling & Software*, 48:37–48, 2013.

[120] Sebastian Achter, Melania Borit, Edmund Chattoe-Brown, and Peer-Olaf Siebers. RAT-RS: a reporting standard for improving the documentation of data use in agent-based modelling. *International Journal of Social Research Methodology*, 25(4):517–540, 2022. doi: 10.1080/13645579.2022.2049511.

[121] Nicole J. Saam. The Users' Judgements—The Stakeholder Approach to Simulation Validation. In Claus Beisbart and Nicole J. Saam, editors, *Computer Simulation Validation: Fundamental Concepts, Methodological Frameworks, and Philosophical Perspectives*, pages 405–431, Cham, 2019. Springer International Publishing.

[122] Sabrina Scherer, Maria A. Wimmer, and Suvad Markisic. Bridging narrative scenario texts and formal policy modeling through conceptual policy modeling. *Artificial Intelligence and Law*, 21(4):455–484, 2013.

[123] Ulf Lotzmann and Martin Neumann. Simulation for Interpretation: A Methodology for Growing Virtual Cultures. *Journal of Artificial Societies and Social Simulation*, 20(3):13, 2017.

[124] Chris J. Topping, Toke T. Høye, and Carsten Riis Olesen. Opening the black box—development, testing and documentation of a mechanistically rich agent-based model. *Ecological Modelling*, 221(2):245–255, 2010.

[125] Peer-Olaf Siebers and Franziska Klügl. *What Software Engineering Has to Offer to Agent-Based Social Simulation*, pages 81–117. Springer International Publishing, Cham, 2017.

[126] JASSS: The Journal of Artificial Societies and Social Simulation. https://www.jasss.org/JASSS.html. Last Accessed: 21/09/2022.

[127] Volker Grimm, Uta Berger, Donald L. DeAngelis, J. Gary Polhill, Jarl Giske, and Steven F. Railsback. The ODD protocol: A review and first update. *Ecological Modelling*, 221(23):2760–2768, 2010.

[128] Volker Grimm, Steven F. Railsback, Christian E. Vincenot, Uta Berger, Cara Gallagher, Donald L. DeAngelis, Bruce Edmonds, Jiaqi Ge, Jarl Giske, J. Groeneveld, uuml, rgen, Alice S. A. Johnston, Alexander Milles, Jacob Nabe-Nielsen, Gary Polhill, Viktoriia Radchuk, Rohw, auml, Marie-Sophie der, Richard A. Stillman, Jan C. Thiele, Ayll, oacute, and Daniel n. The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism. *Journal of Artificial Societies and Social Simulation*, 23(2):7, 2020.

[129] Uri Wilensky and William Rand. Making models match: Replicating an agent-based model. *Journal of Artificial Societies and Social Simulation*, 10(4):2, 2007.

[130] Uri Wilensky. NetLogo Home Page. http://ccl.northwestern.edu/netlogo/. Year of Access: 2019.

[131] JetBrains. MPS:Meta Programming System Version 2021.2. https://www.jetbrains.com/mps/, 2021.

[132] A Conversation with Bertrand Russell (1952). https://www.youtube.com/watch ?v=xL_sMXfzzyA. Last Accessed: January 2023.

[133] Judith A. Hall and Mark L. Knapp. *Nonverbal communication*, volume v. 2. De Gruyter Mouton, 2013.

[134] Jörg Meibauer. The Linguistics of Lying. *Annual Review of Linguistics*, 4(1):357–375, 2018.

[135] Jeremy C. Stein. Cheap Talk and the Fed: A Theory of Imprecise Policy Announcments. *Amercian Economic Review*, 79(Mar.):32–42, 1989.

[136] Tuna Kuyucu. Law, Property and Ambiguity: The Uses and Abuses of Legal Ambiguity in Remaking Istanbul's Informal Settlements. *International Journal of Urban and Regional Research*, 38(2):609–627, 2014.

[137] Jessy Nassar and Nora Stel. Lebanon's response to the Syrian refugee crisis – Institutional ambiguity as a governance strategy. *Political Geography*, 70:44–54, 2019.

[138] Gillian K Hadfield. Weighing the value of vagueness: An economic perspective on precision in the law. *Cal L. Rev.*, 82:541, 1994.

[139] Martin Neumann, Andreas Braun, Eva-Maria Heinke, Mehdi Saqalli, and Armano Srbljinovic. Challenges in Modelling Social Conflicts: Grappling with Polysemy. *Journal of Artificial Societies and Social Simulation*, 14(3):9, 2011.

[140] Amineh Ghorbani, Pieter Bots, Virginia Dignum, and Gerard Dijkema. MAIA: a Framework for Developing Agent-Based Social Simulations. *Journal of Artificial Societies and Social Simulation*, 16(2):9, 2013.

Paper A

Generating Executable Code from High-Level Social or Socio-Ecological Model Descriptions

Themis Dimitra Xanthopoulou, Andreas Prinz, F. LeRon Shults

The presented version is the version published at the AURA - Agder University Research Archive. This paper has been published as:

Themis D. Xanthopoulou, Andreas Prinz, and F. LeRon Shults. Generating exe- cutable code from high-level social or socio-ecological model descriptions. In Pau Fonseca i Casas, Maria-Ribera Sancho, and Edel Sherratt, editors, System Analysis and Modeling. Languages, Methods, and Tools for Industry 4.0, pages 150–162. Springer International Publishing, 2019

Generating Executable Code from High-level Formal Description of Social or Socio-Ecological Models*

Themis Dimitra Xanthopoulou[1][0000−0003−2914−0472], Andreas Prinz[1][1111−2222−3333−4444], and Fount LeRon Shults[2][2222−−3333−4444−5555]

[1] University of Agder, Jon Lilletuns vei 9, 4879 Grimstad, Norway
themis.d.xanthopoulou@uia.no
https://www.uia.no/en/kk/profil/themisdx
[2] University of Agder, Universitetsveien 19, 4630 Kristiansand, Norway

Abstract. Agent-Based Modelling has been used for social simulation because of the several benefits it entails, including the capacity to improve conceptual clarity, enhance scientific understanding of complex phenomena, and contribute policy-relevant insights through simulation experiments. Social models are often constructed by inter-disciplinary teams that include subject-matter experts with no programming skills. These experts are typically involved in the creation of the conceptual model, but not the verification or validation of the simulation model. The Overview, Design concepts, and Details (ODD) protocol has emerged as a way of presenting a model at a high level of abstraction and as an effort towards reproducibility of Agent Based Models (ABMs). However, this popular protocol does not improve the involvement of experts because it is typically written after a model has been completed. This paper reverses the process and provides non-programming experts with a user-friendly and extensible tool called ODD2ABM for creating and altering models on their own. This is done by formalizing ODD using concepts abstracted from the NetLogo language, enabling users to generate NetLogo code from an ODD description automatically. We verified the ODD2ABM tool with three existing NetLogo models and assessed it with criteria developed by other researchers.

Keywords: Social model · Meta-model · Code generation · Abstraction · Formality · Reproducibility · Verification .

1 Introduction

In recent years there has been a rapid rise in the use of the Agent-Based Modelling [2], which typically involves micro-simulation based on individually (inter)acting sub-models [29]. Distinct from the more generic Multi-Agent Models,

* Supported by the University of Agder

ABMs offer unique capabilities and are widely used by a growing number of scientists and policy professionals [5, 10, 11, 21]. One of the prominent applications of ABMs is in social simulations.

ABMs contribute to conceptual clarity for the ambiguous concepts in social science. For example, most of us immediately grasp the general meaning of the concept of bullying, but it can be difficult to determine precisely the sort of incident that falls into that category. Definitions of bullying that are too vague make it difficult to identify and mitigate the relevant behaviours. The creation of an ABM requires scientists to clarify vague concepts by explicitly linking the concepts to agent behaviours, agent properties and interactions, which are easier to compare to real life situations and more likely to render policy-relevant insights.

Hassam et al. have identified four roles in the development of an ABM: the thematician, the modeller, the computer scientist and the programmer [11]. In the following section we will use the paradigms of the model development process presented in [22] to explain the function of each role. A model is the representation of a real-life "problem entity". The *thematician*, who is the expert in the problem entity, is the person who creates the first conceptual model of the problem entity by providing the context of the ABM, system theories and expert knowledge. The *modeller* transforms the description of the first conceptual model into a more formal conceptual model or a simulation model specification, the *computer scientist* finds an executable approximation, and the *programmer* implements the simulation model using a selected programming language and platform. Rarely can one researcher cover all these roles. Moreover, complex models usually require perspectives from different disciplines and therefore more than one thematicians. These are the reasons why ABMs are typically constructed by multidisciplinary teams.

These teams face several challenges. First of all, communication of the conceptual model among such diverse researchers can be difficult [11, 21, 23]. Since the product of each development step depends on the individual conceptual understanding, even with the same *thematician*, different teams may come up with dissimilar simulation models. This dissimilarity hinders reproducibility of results, which is one of the pillars of the *Scientific method*. One of the ways to ensure reproducibility is to perform verification from one step to another. Sargent [22] specifies two types of verification: the *specification verification* that takes place between the conceptual model and the simulation model specification and the *implementation verification* that takes place between the conceptual model specification and the simulation model. The complicating issue is that subject-matter experts, who form conceptual model by abstracting their understanding of the real-life problem entity, are not usually skilled in modelling and computer programming and cannot perform the verifications. They typically find the executable code obscure [4]. The definition of the model becomes 'hidden' in the simulation platform and cannot be perceived, validated or changed by the experts [4, 11]. Nevertheless, researchers urge the involvement of experts in the creation of the model to ensure the validity of the simulation model specification and simulation model [4–6, 11, 21].

Consequently, we want to solve the following problem with this paper: How can we make ABMs more accessible to subject-matter experts to ensure verification and to enable validation of the simulation models? A possible approach favours building blocks that could enable non-programming experts to develop and modify their ABMs [10]. Continuing with this thought, we want to create a domain-specific language (DSL) and an associated tool allowing subject-matter experts to create and change the simulations models by changing their descriptions of conceptual models. Keeping this perspective in mind, we focus more on ease of use than efficiency in code generation.

This paper is structured as follows: Section 2 introduces ODD, NetLogo, and DSL and provides an overview of related work. Section 3.1 describes the process we followed to build each meta-model component. Section 4 presents our results and Section 5 discusses the quality of the outcome. Finally, Section 6 summarizes the paper. Throughout the paper, we illustrate the steps of the methodology with the Wolf Sheep Simple 5 model [25].

2 Background and Related Work

2.1 ODD

Many scholars in the Agent-Based Modelling community have adopted the Overview, Design concepts, and Details (ODD) protocol to further discussions among multidisciplinary researchers about their models. ODD emerged as an effort to "make model descriptions more understandable and complete, thereby making ABMs less subject to criticism for being irreproducible" as Grimm et al. state [7]. If we want to track the ODD description in the model development phases we would identify it as the simulation model specification or at the very least an intermediate step between the conceptual model and the simulation model specification. The protocol has seven thematic sections: "Purpose", "Entities, State Variables and Scales", "Process Overview and Scheduling", "Design Concepts", "Initialisation", "Input Data" and "Sub- models" [8]. Each section contains questions to guide modellers in the provision of related model details.

Figure 1, for example, shows the questions for the "Entities, State Variables and Scales" element of a model. The modeller provides the model definition by answering all the questions. The emerging document with the ODD specifications can be quite large, depending on the model it describes. The answers appear informally, and one can portray the protocol as a group of informal entities. The questions attempt to cover different perspectives of the conceptual model so that all researchers have eventually the same impression of what the model entails. Nowadays, many journals consider the ODD protocol as a prerequisite for the publication of an ABM model. Although this is a big step towards verification, validation and reproducibility of simulation models, the informal character of the answers allow ambiguities in the model description. Platforms such as the

Fig. 1. Informal ODD: Questions and Specification

ODD Element	Questions	Specification
Entity, State Variables, and Scales	What kind of entities are in the model? Do they represent managers, voters, landowners, firms or something else? By what state variables or attributes are these entities characterised? What are the temporal and spatial resolutions and extents of the model?	The agents represent sheep and wolves, and the environment is grassland that they inhabit. Both wolves and sheep have energy that they use to move around. On the other hand, the grass contains energy. The space represents a grassland and the time is not defined by the modeller, but, given the model dynamics, it should be within the lifetime of a sheep.

"ComSES Network OpenABM" mentioned in [12] enable the upload and sharing of ABMs in terms of executable code, ODD and other descriptions to promote model transparency and reuse and to move further towards a scientific handling of ABMs. There are more than 80 platforms that accommodate ABMs based on different programming languages [3]. SWARM covers a large range of models, MASON exhibits fast computational time and Repast is considered the best choice time-wise and modelling-wise [3, 19]. Unsurprisingly, all of them display shortcomings. For example, Repast is not well documented [19]. Our choice for a low-level language and simulation platform is NetLogo [26]. Uri Wilensky created NetLogo to facilitate the development of Agent-Based Modelling and Simulation. The platform has been widely used in the modelling community, and it is relatively simple for non-programmers to use [10], especially in the construction of ABMs [17]. Not only does NetLogo make it easier to develop a model, but it also provides an interface that facilitates simulations and reduces the amount of time needed to design them. In essence, a person with no modelling experience can explore a NetLogo model on the platform. Finally, NetLogo is an open source software.

2.2 DSLs and MPS

Domain-specific languages (DSLs) help efficient development and artefacts. However, it is not enough to have a DSL; one also needs an appropriate tool to work with the DSL. This is normally accomplished with a meta-tool creating a DSL tool out of a DSL description. Such meta-tool is called language workbench.

To build our tool, which we call ODD2ABM, we selected the Meta Programming System (MPS), a free platform for the creation of DSLs [13]. MPS provides projectional editing, which makes it easier to update a meta-model [9], in our case ODD2ABM. We create an ODD inspired editor so that the experience of importing the specifications resembles the experience of writing the ODD in a text file. A DSL description in MPS is structured in the aspects structure, editor, generator, and constraints. Finally, MPS provides tabular and diagrammatic notations in addition to plain text.

2.3 Related Work

Domain-specific languages together with Model Driven Development (MDD) [4] have often been used to solve problems similar to ours. DSLs aspire to provide the model definition in a high-level language so that experts can understand and modify the (domain-specific) model. MDD aims to automate the processes from the high-level description to the lower-level code. The processes of interest make use of languages with different abstraction levels and usually move from a high-level to a low-level language. The developer defines the transformations from one stage to the other. The end user inputs specifications in the high-level language and the DSL tool (semi)automatically generates the next stage artefact or the final code in the low-level language.

Some researchers have worked on the formalisation of certain aspects of ABMs, such as interactions [15]. Others have built meta-models that specialise within specific domains. The MAIA meta-model (Modelling Agent systems based on Institutional Analysis) covers Social Simulations with Institutional Analysis and semi-automatic code generation [6]. MDA4ABMS merges both DSL and MDD methodologies, but the user needs some modelling experience to handle the high-level language, and the tool does not automatically provide the low-level language artefact [4]. Also, the inclusion of UML (Unified Modelling Language) [18], which is applied in the methodology, has often been discarded by other researchers due to its lack of expressiveness [21]. Similarly, the easyABMS methodology includes UML and does not provide automatic code generation but takes into account all the modelling and simulation phases and can be used for general processes [5]. The meta-model introduced by Santos et al. [21] automatically generates code, and has been evaluated as very efficient; however, it only functions for a particular domain. Finally, adaptations of Multi-Agent methodologies to Agent-Based have been able to establish a common high-level formal language, but these do not include automatic code generation [11].

3 Methodology

This paper aims at creating a DSL and an associated tool for the construction and modification of ABMs. The central idea is that the user will input the conceptual model in the DSL and the tool will automatically generate the simulation model in NetLogo executable code. Apart from advantages related to the creation or modification time of conceptual and simulation models, the method provides build-in verification of the model. The main reason is that there is a deterministic relationship between input (formally described conceptual model) and output (executable code or simulation model). In essence, the verification is the tool itself. The goal is to start from simple models, so that we obtain a proof of concept for our idea, and then extend the work to more complex ABMs.

The tool should be easy to use, extensible, and allow automatic code generation from the model definitions produced by subject-matter experts. Our methodology integrates aspects of MDD and DSL. The DSL ensures user-friendliness and accommodates diverse models, while the MDD is used to provides the code generation into a lower level simulation language. Using MPS as the tool of choice for DSL development enables a focus onto the language description. MPS will generate a neat and efficient DSL tool directly and automatically from the language description.

As ODD already is a DSL for social models, a tool that transforms an ODD to NetLogo would solve our problem. However, ODD is not formal enough for a direct transformation. Still, we argue that instead of creating our DSL from scratch, using the methodologies proposed by [5, 11, 21], we can take advantage of the accumulated experience of researchers that the ODD incorporates and formalise it (i.e. we make the descriptions already in ODD more formal). Some of the advantages of using the protocol as a starting point include its pre-existing structure [14] and its inclusion of Agent-Based Modelling domain concerns that cover the need for the DSLs broadness and extensibility. Skipping the domain analysis and spending less energy to organise ODD, reduces the effort of the DSL development. Although Santos et al. [21] used the ODD protocol to refine the collection of concepts for the domain analysis of their case study, researchers have not yet taken full advantage of it to render ABMs more accessible.

We argue that NetLogo is a good starting point as the simulation language of our tool since it is well documented [1] and accommodates a variety of models (but not large scale ones). Since it is a higher-level language than general-purpose programming languages such as Java, it will be easier to design transformations from NetLogo to Java or similar programming languages in a later phase of our project.

Using this method, the two remaining challenges are: (1) a formalisation of the sections of ODD with important information for the simulation, and (2) a description of the transformation from ODD to NetLogo. The formalisation of ODD is closely related to the user friendliness of ODD2ABM. We want to make our DSL so accessible that it could be used by experts without any programming experience. This ease of use is intended to encourage and enable such users to construct and adapt AMBs without overly relying on computer scientists and coders. Moreover, we want to make our DSL capable of incorporating a broad range of models. Integrated perspectives support robust models such as [16]. In this example, Kuil et al. incorporated a fusion of social and ecological dynamics that managed to explain the decline of the Mayan civilisation [16]. Therefore, it is important to accommodate different types of models and not restrict users to a specific domain.

The two main challenges in the creation of a formal ODD are the repetition of information, and missing information. Missing information is information that is available in the NetLogo code, but that is not present in the ODD. For such information we have to find out whether the information can be generated from other existing information, or whether it must be included into the ODD.Repeated information could be handled by just ignoring the duplicated parts. However, it is important that the formal parts of ODD are reliable and if there is duplicated information, it has to be synchronized. In MPS this problem involves determining the primary place of the information and the creation of a reference to it at all the other uses.

One aspect of the tools friendliness is its capacity to automatically generate executable code, a task which has previously required programming skills. Using MPS, it is straightforward for non-programming experts to run their simulations. Although we have chosen NetLogo as target language for the code generation, there is still a lot of variability in the actual code to be produced. This again might influence the choice of concepts in the DSL, as we would prefer concepts that are easily implemented.

3.1 Meta-model elements

To create ODD2ABM in MPS, we needed to define the structure, constraints, editor, and generation rules of the DSL. ODD itself comes with an editor as shown in the Figure 1. ODD2ABM should use a similar editor reusing existing elements and adding new ones when necessary. It was not clear from the start which formal elements would be needed for ODD. To determine them, we used the following systematic procedure. The procedure organised the input in such a way as to ensure that the ODD user provides the data to cover the specifications. It resulted in a DSL description for structure and constraints. Finally, we created generation rules for automatic code generation.

3.2 Procedure for defining the Meta-model Structure

Collection of the ODD elements and questions We selected the ODD version from Grimm et al. [8], which includes seven ODD elements.

Selection of NetLogo models for concrete model instances and code Since this is the first version of ODD2ABM, we chose to start with simple models from the NetLogo library. The models we used are the Segregation Simple model [24], the Fire simple model [27], and the Wolf Sheep Simple 5 model [25]. For each of the models, we had an ODD description and NetLogo code. In parallel, we consulted the NetLogo dictionary [1] and a chapter specialised on Agent-Based modelling concepts for NetLogo [28]. The dictionary and the ABM concept overview ensured that the the simplicity of the first test models will not comprise the extensibility and capacity for variety of our meta-model.

Match of each element with the corresponding code For each element in the code of the selected models, we attempted to find matching information in the ODD description. For example, for the element "Entities, State variables, and Scales" and the entity "sheep" (see Figure1), we registered the code "breed [sheep a- sheep" and "sheep own [energy]". Questions such as "What are the temporal and spatial resolutions and extensions of the model?" are not semantically significant for the code. Using the final code, we distinguished the ODD elements that produce parts of the code from those that do not.

Identification of the parts of NetLogo code that cannot be extracted from the ODD specifications One example is the NetLogo entities, which can be turtles or patches. Turtles are moving agents, which can have capacities such as interacting with one another and recognising their location. We can think of patches as part of the grid on the simulation space. They cannot move, but they have properties that can change (such as colour). Patches interact with turtles but also with their neighbouring patches. Finally, NetLogo links are connection lines between the moving agents that indicate some sort of relationship. The distinction between the different entities is not visible in an ODD specification. We attempted to distinguish between low-level information that should not be included into ODD from higher-level information that should be included. The method to accomplish this was to formulate the information on the level of ODD. We excluded information not conceptually level and used a low-level way to produce it.

Creation of questions in the ODD language to accommodate code generation To follow up the previous example, we created questions on whether the entity is part of the environment or not. The questions were first captured in a flow diagram. Then they were incorporated in the editor description or editor structure. For example, there is a different place in the editor to define general entities and environmental entities. The same procedure applies for the rest of the model.The answers are the specifications of the meta-model, which enable the code generation. The specifications with no semantic significance for the code generation, such as the Purpose statement (see Figure 3 require an informal textual answer.

Grouping questions that reappear It is possible to locate the same questions in multiple positions of ODD. For example, to define the descriptive characteristics of the environment or the total population or the attribute of an entity we need to input the same type of specifications. There are two reasons for grouping this information: first to reduce the time for the DSL development and second to enhance the visual representation of the specifications by reducing the amount of information in them.

Extraction of the DSL structure from the diagram Figure 2 illustrates part of the concepts of the DSL that relate to the ODD element "Entities, State Variables and Scales". Each user input corresponds to either a DSL concept (for example the concept Entity) or to a DSL concept attribute (for example the attribute label of the concept Entity).

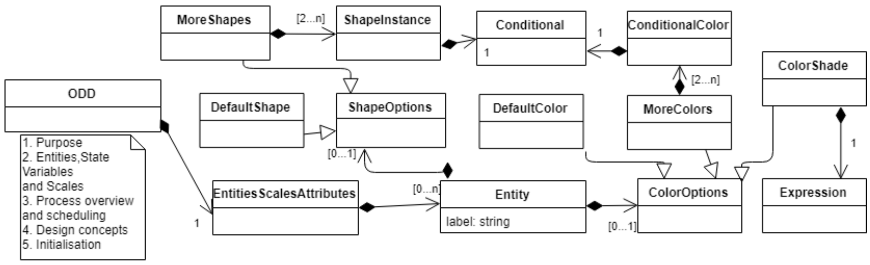


Fig. 2. UML class diagram of Entities, State Variables, and Scales

Registration of constraints that emerge from the structure Finally, we made sure that specifications are meaningful and did not violate common sense. For this, we extracted the informal conditions placed on the concepts and formalize them as MPS constraints. If we take the example of the entity sheep (Figure 3), we read that it contains the attribute "energy". The attribute is of type float. This will determine the next specification, which will be to define the range of values. If energy was of type string, then the editor would request a list of string values. In addition, the type of the variable and the specification "float" will constrain the input for other elements such as the initialisation. Finally, action is possible only for the defined attributes. The type of attribute selected in the specifications of "Entities, State Variables, and Scales" will constrain the value that the user imports in the Initialisation element. For example, the attribute "energy" was defined as "float". If the user were to assign a string value, the editor would not have accepted it. Overall, there are two types of constraints: those associated with the description in the specification and those associated with whether a specification is even available according to previous input.

3.3 Editor

The UML diagrams of the newly formalised ODD structure resemble the diagrams of the informal one in the sense that they expand from the seven elements. The identified questions of step five complement the initial ODD structure. The editor for ODD is providing textual and tabular syntax for the structural elements of ODD. The original look and feel of ODD is kept by putting the elements into text as much as possible and keeping the concepts at the same language level. The editor of MPS provides state-of-the-art content already from the start. Some more advanced features can be added manually. In particular, for the case of ODD it is important that the required input is very specific, except for answers with no semantic significance. For example, we can see in Figure 3, that the editor description is "Color is defined for the entity". The initial text is "Color ¡Press Alt and Enter to choose to include or not include color¿ defined for the entity". When the user presses Alt and Enter the only choices are "is" and "is not". The MPS editor provides auto-complete to help users identify possible

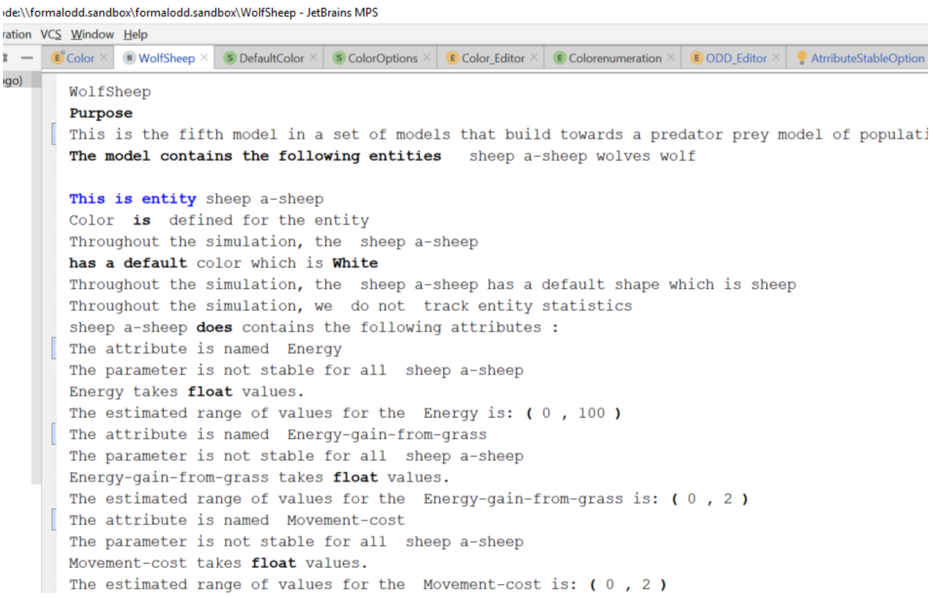


Fig. 3. Formalised ODD in MPS

continuations and lists where we want to define the possible answers, as well as static checks on the fly in order to avoid wrong inputs. For more complex inputs, the editor description indicates and checks the right way to configure the text. Because of the projectional editor poperty of MPS, it is not possible to see questions that are not enabled in the current model. For example, in the previous example, when the user selects "is not" then the part where the user specifies the color method and the specific color choice, currently visible in Figure 3, are not shown. The editor appearance is derived from the questions, which is similar to the corresponding concepts (see Figure 2). Overall, the elements are connected and the full UML diagram shows all the connections.

3.4 Executable Code Generation

Normally, code generation follows the flow of information as given in the ODD structure. Still, depending on the place in the generated code, it might be the case that more information is collected from different parts of the specification. For example, even though we choose to specify whether an entity has a color in the Entities State Variables and Scales element, MPS uses these specifications to generate code in the Initialisation part.

Code generation requires the specification to be statically correct, i.e. all constraints should be satisfied. This is checked already in the editor and signalled to the user.Code generation is normally disabled as long as there are errors in the specification. Vice versa, utmost care was given to make sure that the code generation is successful whenever the specification is statically correct. For example, if we look at Figure **??**, generating the code "sheep own [energy]" requires the user to assign the attribute "energy" to the entity "sheep". If no entity has been defined, it will not be possible to define an attribute for it. Moreover, the label of the attribute has to be defined in order to enable generation of correct code. Sometimes code is only generated under specific circumstances, which explains why some rows in the "Generated Code" column are empty.

We can identify two categories of code in NetLogo: the code for the model definition and the code for the simulation setup. In our DSL, we do not differentiate between the two types. To the best of our knowledge, all models in NetLogo contain the Setup and the Go buttons in the NetLogo interface. Using these buttons, the modeller can restore the entities to the initial condition and start/stop the simulation. Even if the particular naming of the buttons ("Setup" and "Go") is not mandatory, the code does not run without the existence and pressing of a button in the user interface. All procedures are initiated when called directly or indirectly from one NetLogo button. Therefore, we choose the generation "Setup" and "Go" buttons by default in the code generation from the DSL.

4 Solution

We performed the eight-step procedure outlined in Section 3.2 for the four ODD elements: "Purpose", "Entity, State Variables, and Scales", "Process Overview and Scheduling", and "Initialisation". For each one, we created a Diagram that summarised the meta-model specifications and extracted the UML diagram. We verified the tool for the first 3 elements using the 3 NetLogo models (the Wolf-Sheep Simple 5, Segregation and Fire model). Figure 2 shows part of the structure for the Entities element and Figure 3 shows the editor from which the code "breed [sheep a- sheep" and "sheep own [energy]" was generated. The editor text serves as the ODD document.

5 Evaluation

5.1 Expressivity and Extension

We created our DSL based on relatively simple models and so cannot guarantee that ODD2ABM covers the range of model specifications that are needed for social simulation. However, the concepts are carefully chosen to cover a broad range of application such that there is at least a major range of possible specifications. In the future, we will validate the DSL with more complex ABMs and introduce more specifications. For example, we plan to further develop several aspects of the tool such as how it deals with interactions and relationships among agents marked in NetLogo with links, and add the possibility of importing data from files external to the platform. The extensibility of our DSL is ensured by MPS and the conceptual framework we adopted.

5.2 ODD and Experts

The ODD protocol gave us the opportunity to enrich it as a DSL with some modifications without losing its accessibility for users with limited programming skills. The question remains whether the level of the language is high enough for thematicians to engage with it. Part of the concern lies in the fact that the original ODD targets modellers. We all use models (in a general sense) in our everyday lives. However, the concepts employed in the Agent-Based Modelling community, such as entities and attributes, may not be intuitively clear to all users. Therefore, even if the tool is very effective for modellers, experts not familiar with the ODD language may face difficulties in its implementation. In a next stage of development we will evaluate the usefulness of ODD2ABM.

6 Summary and Future Steps

The ODD2ABM tool described in this paper serves as a proof of concept for a methodology that incorporates DSL and MDD, uses the MPS platform, enables experts to create, modify their ABMs and provides a new way to ensure reproducibility of results. During the construction of this tool, we were careful to ensure its user-friendliness and extensibility. We selected the ODD protocol as the basis for our DSL and NetLogo as our low-level language. The resulting DSL is original in its capabilities and properties as it accommodates a large range of modelling themes and enables automatic executable code generation. We plan to broaden ODD2ABM so that it allows more freedom in model creation. A next step would be to survey experts from different disciplines to discover whether our formalisation of the ODD protocol needs to be abstracted further in a process such as the one defined in [20].

References

1. Netlogo dictionary, https://ccl.northwestern.edu/netlogo/docs/dictionary.html
2. Abar, S., Theodoropoulos, G.K., Lemarinier, P., O'Hare, G.M.P.: Agent based modelling and simulation tools: A review of the state-of-art software. Computer Science Review 24, 13–33 (2017)
3. Abar, S., Theodoropoulos, G.K., Lemarinier, P., OHare, G.M.: Agent based modelling and simulation tools: A review of the state-of-art software. Computer Science Review 24, 13 – 33 (2017), http://www.sciencedirect.com/science/article/pii/S1574013716301198
4. Garro, A., Parisi, F., Russo, W.: A Process Based on the Model-Driven Architecture to Enable the Definition of Platform-Independent Simulation Models, pp. 113–129. Springer Berlin Heidelberg, Berlin, Heidelberg (2013), https://doi.org/10.1007/978-3-642-34336-0_8
5. Garro, A., Russo, W.: Easyabms: A domain-expert oriented methodology for agent-based modeling and simulation. Simulation Modelling Practice and Theory 18(10), 1453 – 1467 (2010), http://www.sciencedirect.com/science/article/pii/S1569190X10000717, simulation-based Design and Evaluation of Multi-Agent Systems

6. Ghorbani, A., Bots, P., Dignum, V., Dijkema, G.: Maia: a framework for developing agent-based social simulations. Journal of Artificial Societies and Social Simulation 16(2), 9 (2013), http://jasss.soc.surrey.ac.uk/16/2/9.html
7. Grimm, V., Berger, U., DeAngelis, D.L., Polhill, J.G., Giske, J., Railsback, S.F.: The odd protocol: A review and first update. Ecological Modelling 221(23), 2760 – 2768 (2010), http://www.sciencedirect.com/science/article/pii/S030438001000414X
8. Grimm, V., Polhill, G., Touza, J.: Documenting Social Simulation Models: The ODD Protocol as a Standard, pp. 117–133. Springer Berlin Heidelberg, Berlin, Heidelberg (2013), https://doi.org/10.1007/978-3-540-93813-2_7
9. Guttormsen, S., Prinz, A., Gjøsæter, T.: Consistent projectional text editors. In: MODELSWARD. pp. 515–522 (2017)
10. Hamill, L.: Agent-based modelling: The next 15 years. Journal of Artificial Societies and Social Simulation 13(4), 11 (2010), http://jasss.soc.surrey.ac.uk/13/4/7.html
11. Hassan, S., Fuentes-Fernández, R., Galán, J.M., López-Paredes, A., Pavón, J.: Reducing the modeling gap: On the use of metamodels in agent-based simulation. In: 6th conference of the european social simulation association (ESSA 2009). pp. 1–13 (2009)
12. Janssen, M.A., Alessa, L.N., Barton, M., Bergin, S., Lee, A.: Towards a community framework for agent-based modelling. Journal of Artificial Societies and Social Simulation 11(2), 6 (2008), http://jasss.soc.surrey.ac.uk/11/2/6.html
13. JetBrains: MPS Meta Programming System, https://www.jetbrains.com/mps/
14. Klügl, F., Davidsson, P.: Amason: Abstract meta-model for agent-based simulation. In: Klusch, M., Thimm, M., Paprzycki, M. (eds.) Multiagent System Technologies. pp. 101–114. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
15. Kubera, Y., Mathieu, P., Picault, S.: Interaction-oriented agent simulations: From theory to implementation. In: Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence. pp. 383–387. IOS Press, Amsterdam, The Netherlands, The Netherlands (2008), http://dl.acm.org/citation.cfm?id=1567281.1567367
16. Kuil, L., Carr, G., Viglione, A., Prskawetz, A., Blschl, G.: Conceptualizing socio-hydrological drought processes: The case of the maya collapse. Water resources research 52(8), 6222–6242 (2016)
17. Lytinen, S.L., Railsback, S.F.: The evolution of agent-based simulation platforms: A review of netlogo 5.0 and relogo. In: Proceedings of the fourth international symposium on agent-based modeling and simulation (21st European Meeting on Cybernetics and Systems Research [EMCSR 2012 (2012)
18. OMG: Unified Modeling Language: Infrastructure version 2.4.1 (OMG Document formal/2011-08-05). OMG Document, Published by Object Management Group, http://www.omg.org (August 2011)
19. Railsback, S.F., Lytinen, S.L., Jackson, S.K.: Agent-based simulation platforms: Review and development recommendations. SIMULATION 82(9), 609–623 (2006), https://doi.org/10.1177/0037549706073695
20. Santos, F., Nunes, I., Bazzan, A.L.C.: Supporting the development of agent-based simulations: A dsl for environment modeling. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC). vol. 1, pp. 170–179 (July 2017)
21. Santos, F., Nunes, I., Bazzan, A.L.: Model-driven agent-based simulation development: A modeling language and empirical evaluation in the adaptive traffic signal control domain. Simulation Modelling Practice and Theory 83, 162 – 187

(2018), http://www.sciencedirect.com/science/article/pii/S1569190X17301673, agent-based Modelling and Simulation
22. Sargent, R.G.: Verification and validation of simulation models. In: Proceedings of the 2010 Winter Simulation Conference. pp. 166–183 (Dec 2010)
23. Wildman, W.J., Fishwick, P.A., Shults, F.L.: Teaching at the intersection of simulation and the humanities. In: 2017 Winter Simulation Conference, WSC 2017, Las Vegas, NV, USA, December 3-6, 2017. pp. 4162–4174 (2017), https://doi.org/10.1109/WSC.2017.8248136
24. Wilensky, U.: Netlogo seggregation model. Report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1997), http://ccl.northwestern.edu/netlogo/models/Fire
25. Wilensky, U.: Netlogo wolf sheep predation model. Report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1997), http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation
26. Wilensky, U.: Netlogo. Report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999), http://ccl.northwestern.edu/netlogo/
27. Wilensky, U.: Netlogo fire model. Report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (2006), http://ccl.northwestern.edu/netlogo/models/Fire
28. Wilensky, U., Rand, W.: The Components of Agent-Based Modeling, pp. 203–282. The MIT Press, Cambridge, Massachusetts ;, 1st edn. (2015)
29. Wurzer, G., Kowarik, K., Reschreiter, H.: Agent-based modeling and simulation in archaeology. Springer (2015)

Paper B

Modeling Marginalization: Emergence, Social Physics, and Social Ethics of Bullying

Themis Dimitra Xanthopoulou, Ivan Punga-Gonzalez, F. LeRon Shults, Andreas Prinz

The presented version is the version published at the AURA - Agder University Research Archive with some corrections in the References. This paper has been published as:

Themis D. Xanthopoulou, Ivan Puga-Gonzalez, F. LeRon Shults and Andreas Prinz, Modeling Marginalization: Emergence, Social Physics, and Social Ethics of Bullying, In 2020 Spring Simulation Conference (SpringSim), Fairfax, VA, USA, 2020 pp. 1-12. doi: 10.22360/SpringSim.2020.HSAA.005

MODELING MARGINALIZATION: EMERGENCE, SOCIAL PHYSICS, AND SOCIAL ETHICS OF BULLYING

Themis D. Xanthopoulou
Andreas Prinz
Department of Information and Communication Technology
University of Agder
Jon Lilletuns vei 9
Grimstad, Aust-Agder, Norway
themis.d.xanthopoulou@uia.no
andreas.prinz@uia.no

Ivan Puga-Gonzalez
F.LeRon Shults
Department of Global Development and Planning
University of Agder
Universitetsveien 19
Kristiansand, Vest-Agder
ivan.puga-gonzalez@uia.no
leron.shults@uia.no

ABSTRACT
In this paper, we outline the construction and initial simulation experiment results of the Marginalization model (MARG). We experiment under different group parameters because the theoretical paradigm we follow views bullying as a result of social processes. Our primary research question explores the possibility of bullying emergence as agents select interaction partners in a university setting. Based on the simulated process, our results take indications of the stress of marginalization in a student group as a proxy for emergent marginalization. MARG simulates two types of interactions between pairs of students: forced and hang-out interactions. In the latter, students decide whether to interact based on individual preferences formed by social norms and individual tolerance related to those norms. The emergence of intensified marginalization from MARG processes leads to some ethical considerations and provides ground for discussions concerning suitable interventions.
Keywords: marginalization, bullying, social simulation, ethics

1 INTRODUCTION

A recent report by Ingrid Lund found that 9% of students at universities and colleges in Norway have experienced bullying (Lund 2017). Lund's report attracted a lot of attention, and a working group at the University of Agder was founded by the rector to address the issue. The strong reaction comes as no surprise since bullying is linked with mental health problems and violence (Blood and Blood 2016; Wolke and Lereya 2015). In addition to health issues, students who are bullied are more likely to drop out of school (Cornell et al. 2013). Notwithstanding the ever-growing body of research on bullying, there is no concrete explanation of how bullying emerges (Turner et al. 2015). Most people would agree that it is wrong and unethical to bully someone. But is it possible for this "unethical" act to emerge from an ethically acceptable action such as choosing an interaction partner? In this paper, we explore in an agent-based computer model the effect of University students' partner choices on bullying.

2 THEORIES AND METHODS OF BULLYING

Bullying is a concept employed to describe a wide range of behaviors (Cohen and Brooks 2018), and currently, there is no consensus among researchers for the definition (Turner et al. 2015). The large number of interpretations make it very difficult to impossible to integrate all definitions into one model consistently. In addition to the myriad of behaviors, there are currently two paradigms for understanding bullying (Schott and Søndergaard 2014). The first and older one focuses on the individual traits of involved parties and the second one on existing social dynamics as the force that drives bullying. The functional difference between the two is that the first paradigm places responsibility on the individual while the second one on the interplay of the dynamics. Several studies support the idea that there are more than individual traits that play a role in bullying. Salmivalli's work highlighted the group character of bullying by introducing the roles of bystanders, defenders of the victim, and supporters of the bully (Salmivalli et al. 1996). In addition, Paluck et al.' s study indicated the importance of changing norms to reduce conflict in schools (Paluck, Shepherd, and Aronow 2016). Finally, the emergence of a socio-ecological model where different norms relate to the bullying phenomenon supports the adoption of the second paradigm to explain bullying (Espelage and Swearer 2010). Based on these findings, we approach bullying with the second paradigm. Based on a popular definition within this paradigm, we operationalize bullying as the "intensification of the processes of marginalization that occur in the context of dynamics of inclusion-exclusion, which shape groups. Bullying happens when physical, social, or symbolic exclusion becomes extreme, regardless of whether such exclusion is experienced or intended" (Schott and Søndergaard 2014).

The context of our paper is bullying at the University. To be more specific, we are interested in the exclusion of students from interactions that take place in their free time but within the space of the University. Some examples would be interactions during recess and group study. Exclusion is another ambiguous concept (Peace 2001) with multiple dimensions (Mathieson et al. 2008), and we need to define it further. We focus on exclusion related to communication with peers in the educational environment. We acknowledge three types of interactions: positive, negative, and refused interactions. We view instances of refused and negative interactions as instances of exclusion. For our paper, we consider marginalization as the culmination of exclusion events. Marginalization is expressed for an individual when she has a large ratio of negative or refused interactions or when, on average, the interaction partners are not attracted to the individual and do not want to interact. We define the "intensified" component by using thresholds.

We want to understand if a process such as the selection of interaction partners, which is not considered unethical, could lead to bullying, as defined in the previous paragraph. Based on the theory developed by Thibault and Kelly, the result of the interactions is based on the compatibility of each interaction partner's characteristics (Thibaut and Kelley 1959). Compatibility implies the existence of a sort of evaluation in the characteristics of the interaction partner. People choose interaction partners based on the outcomes of previous interactions. In our paper, we view compatibility as a matter of individual preference. Different cultures favor different types of behaviors (Hofstede, Hofstede, and Minkov 2010), and individual preferences are formed within these contexts. We acknowledge that our approach is limited in tracking the manifested bullying in a case study because it includes only a related process and that it will most probably only identify a tendency towards bullying. On the other hand, the limitations do not prevent us from reflecting ethically on the emergence of bullying.

We chose the Agent-Based Modeling (ABM) methodology to explore bullying dynamics in relation to the selection of interaction partners. ABMs have been used before to address bullying but mostly based on predefined roles in bullying for the agents and individual perspectives. Therefore, previous work links to the framework of the first paradigm for explaining bullying. Tseng et al. created an ABM model of bullying using Social Impact Theory and defined student roles as the victim, the bully, and the bystanders (Tseng et al. 2014). Thawiworadilok et al. used an ABM based on game theory (prisoner dilemma game) and assigned with probability student roles to test the effect of the victim's compliance with the bully (Thawiworadilok, Songhori, and Terano 2017). In contrast, we focus on the second paradigm and shift the focus onto the group perspective. Closer to the second paradigm with regards to non-predefined roles, with results still on the individual level, is the model by (Maeda, Anezaki, and Takeuchi 2006). They identify students with a set of values to portray their interests and simulate homogenization processes to understand which students will get excluded and thus have higher chances of being bullied. They showed that students with rare values or interests are more likely to be left out in interactions. In our model, we do not include homogenization processes because we consider them a defense mechanism to counteract intense marginalization. Our main question is, "Can our model produce intensified marginalization as a result of the social process of selecting interaction partners based on individual preferences formed by social norms?"

3 MARG: THE BULLYING MODEL

3.1 Model Overview

MARG is an agent-based model written in Netlogo version 6.1.1 (available here: https://github.com/themisdx/MARG https://github.com/themisdx/MARG) designed to represent the interactions that occur in a university setting among a group of students (the number of students is a model variable called "num-students"), without previous knowledge of each other. Each time step represents a 'day' and therefore the simulation last 100 number of time steps, resembling thus a whole university semester. The most important variables are shown in Table 1 and Table 2. At each time step, agents have two types of interactions: classroom interactions and hang-out interactions. All interactions are dyadic. During classroom interactions, agents cannot refuse to start an interaction with others; therefore, we call these forced interactions. In contrast, during hang-out interactions, agents may refuse to start an interaction with another agent, and thus we call these free interactions. At each time step and for each type of interaction, agents are paired N times in randomly selected dyads, with N = (num-students – 1)/2.

Interactions result in positive or negative evaluations. These evaluations are based on the comparison between the interaction partner's characteristics and 'ideal' values (see below for details). For every other agent, agents have a variable called "attraction" that represents their likeness to the other agent and thus their disposition to interact with it. This variable is unidirectional, which means that two agents may have a dissimilar attraction towards each other. When the value is 1, an agent feels maximum attraction for that specific agent, and when it is 0, the agent feels minimum attraction.

An important element of hang-out interactions is the element of choice. During hang-out interactions (free interactions), after an agent has been randomly paired with another, it first decides whether it wants to interact or not with its partner by comparing its attraction link towards the other agent with a random number between 0 and 1. If the attraction is higher than the random number, the agent proceeds with the interaction (note that during forced -classroom- interactions, the first agent always starts an interaction regardless of the value of the attraction link towards its partner). Then, in the hang-out interactions, the other agent must decide whether to proceed with the interaction on the basis of the same decision rule. In the hang-out interactions, if the first agent refuses, the second one is unaware of the decision. On the other hand, when the first agent starts the interaction and the interaction partner rejects the interaction, the first agent will acknowledge this as a refused interaction and will increment its #refusedinteractions by 1. In both forced and free interactions, if an interaction is positive, then the agent increases its attraction towards the interaction partner by the amount of "Attraction_change" and thus the likelihood of interacting with that agent in the future. If, on the other hand, the interaction is negative, then the agent decreases its attraction link by the same amount and thus the likelihood of interacting with that agent in the future.

Table 1: Simulation Parameters.

Group variables	**Description**	**Range**	**Increment**
Num-students	Number of students in the simulation	10-50	10
Num-internal-characteristics	Number of internal characteristics per agent	Fixed: 10	NA
Num-external-characteristics	Number of external characteristics per agent	Fixed: 10	NA
Average_char	Average value "Average_char" and standard deviation "Stdev_char" of the truncated (min 0 and max 1) normal distribution from which values are drawn for the agents' characteristics	Fixed: 0.5	NA
Stdev_char		0.1-0.5	0.1
Attraction_change	Change in the attraction link of an interaction partner after a positive or negative interaction.	0.02-0.1	0.02
Attitude	Initial attraction link value towards the other interaction partners.	Fixed: 0.5	NA
Max_judg	Maximum value of the uniform distribution from which values of tolerance are drawn.	0.1-0.5	0.1
Charlearned	Number of internal-characteristics an agent learns about an interaction partner during an interaction	Fixed: 1	NA
In_ideal_chars	Optimal cultural value of the internal characteristics (shared for all agents).	Scenario 1: 0.5 Scenario 2: 1.0	NA NA
Ex_ideal_chars	Optimal cultural value of the external characteristics (shared for all agents).	Scenario 1: 0.5 Scenario 2: 1.0	NA NA

At initialization, Attraction towards others is set to the value of "Attitude". In our simulations, Attitude equals to 0.5, so that all agents have the same chance of accepting/rejecting an interacting. Further, the decision rule ensures that agents will have a chance to interact with those they will consider not so attractive and reject interactions from those whom they will feel a strong attraction to.

3.2 Characteristics of Agents and Learning

Each agent is endowed with a set of external and internal characteristics. "External-characteristics" represent conspicuous features of persons such as physical appearance; "internal-characteristics" represent more personal features such as personal interests, way of thinking, etc. which are only known through communication. "Num-internal-characteristics" and "num-external-characteristics" allow the manipulation of the number of characteristics in the model. The values of the external and internal characteristics among agents are heterogeneous. Besides these characteristics, agents are also provided with an 'ideal' set of external and internal values. The values are controlled by the variables "In_ideal_chars" and "Ex_ideal_chars" This set of 'ideal' values is the same among all agents and represents the 'cultural' traits, affected by cultural norms, agents look for when interacting with others (see Table 1). The closer the values of the extrinsic and intrinsic characteristics of an agent to the ideal values, the higher the likelihood of being liked by others. However, cultural norms are not the only factor in evaluating the interaction partner positively. Each agent is assigned its own "tolerance" variable. The combination of the ideal and the tolerance value form the individual preference (see Table 2). The higher the tolerance, the wider the range of the individual preference, the less the effect of norms in liking another agent.

At the beginning of the simulations, the agents only know the external characteristics of others; internal characteristics are learned during interactions. The number of internal-characteristics learned during each interaction is expressed by the variable "char-learned", and the characteristics themselves each agent knows for one other agent are stored in an array called "known-indices".

3.3 Evaluations during Interactions

Whether an interaction is positive or negative depends on the evaluation of the agent's characteristics values (extrinsic and intrinsic) against those of the 'ideal' set of characteristics. If the value of the agent's characteristic falls within the individual preference of the evaluator, then the evaluation is positive (value 1), and otherwise, it is negative (value -1). As a result, the higher the tolerance of an agent, the more likely it is to evaluate another agent positively and vice versa. The agent evaluates all extrinsic characteristics and known intrinsic characteristics of the interaction partner. If the sum of these evaluations is positive, then the interaction is positive, and the agent will register +1 #positiveinteractions. In the opposite case, the agent will register +1 #negativeinteractions. The consequence of learning more characteristics about the interaction partners is that the evaluations change as time progresses.

3.4 Experimental Set-Up

Table 1 shows the parameter space for our simulation. All simulation parameters are group parameters due to the adoption of the second paradigm described in Section 2. For the experiments, we manipulate the group parameters, and they form the agent parameters shown in Table 2. The method of generation of the agent parameters is also described in Table 2. We ran simulations under two different scenarios varying four parameters (shown in Table 1). For Scenario 1 or 2 and a combination of parameter set (n=625 or 5^4), we ran five replications (to ensure the reliability of the results), leading to a total of 6250 simulations. Further, based on the values of Max_jud and Stdev_char (Table 1), we categorize simulations into low/medium/high tolerance and diversity, as shown in Table 3.

3.5 Data Collection, Bullying Metrics, and Data Analysis

For each agent, we collected the variables #positiveinteractions, #negativeinteractions, #refusedinteractions, and the average attraction value the other agents felt towards it (calculated in variable "average_attraction_in") at the end of the simulations (see Table 2). With the data collected and taking the perspective of the second paradigm, which relates bullying to social dynamics, we measured bullying as the intensified marginalization 'experienced' by an agent calculated in two different ways:

Metric 1. *"Marginalized?" was based on the exclusion index. When the index passes the threshold of 0.8, meaning that more than 80% of the interactions of the agent are negative, then Exclusion? was set to true. The logic behind the exclusion index is to capture the number of exclusion experiences perceived by the agent in relation to all its interactions. The threshold represents the intensified marginalization. The exclusion index is shown in equation (1):*

$$\text{Exclusion index} = \frac{\#NegativeInteractions + \#RefusedInteractions}{\#NegativeInteractions + \#RefusedInteractions + \#PositiveInteractions}. \quad (1)$$

Metric 2. *"Marginalized_attraction?" was based on the average_attraction_in: when the value of this metric is below 0.2, meaning that on average other agents reject an interaction 8 out of 10 times with the target agent, then Marginalized_attraction? was set to true. This metric represents the exclusion received from others. The threshold represents the intensified marginalization.*

Table 2: Agents' variables and Data Collection variables.

Agent Variables	Description
Tolerance	Individual value assigned to each agent from a uniform distribution with min = 0 and max = Max_jud
Internal-characteristics	The intrinsic characteristics of the agent. Values range between 0 and 1 and are drawn from a normal distribution with average = Average_char and standard deviation = Stdev_char
External-characteristics	The extrinsic characteristics of the agent. Values range between 0 and 1 and are drawn from a normal distribution with average = Average_char and standard deviation = Stdev_char
Individual preference	Range of preference of a characteristic = ideal value ± tolerance
Attraction	A variable representing how much an agent 'likes' another one; values range from 0 (low attraction) to 1 (high attraction). Each agent has one parameter value for each other agent.
known-indices	The exact internal-characteristics an agent knows from another one at a given time step in the simulation for each agent. Each agent has one array of values for each other agent.
Data Collection Variables	**Description**
#positiveinteractions	The total positive interactions the agent had in the simulation

#negativeinteractions	The total negative interactions the agent had in the simulation
#refusedinteractions	The total refused interactions the agent had in the simulation
Average_attraction_in	The average value of incoming attraction of all other agents towards this agent
Students_ marginalized	The percentage of agents excluded in a specific simulation based on a threshold value and the agents' exclusion index
Students_ marginalized_attraction	The percentage of agents excluded in a specific simulation based on a threshold value and the agents' average_attraction_in

These metrics represent two different aspects of exclusion that do not necessarily coincide. Metric 1 represents how a specific agent perceives exclusion from others, and metric 2 represents the perception of the group (average attraction) towards the specific agent. Hence, there may be cases in which a highly tolerant agent perceives positively most of its interactions (because it evaluates others positively), yet itself is not liked and gets often refused interactions from others. In this case, metric 1 will not indicate a lot of negative interactions, while metric 2 will a marginalization signal. On the contrary, it can happen that an intolerant agent dislikes everybody and isolates itself (by refusing to interact with others) even though it is liked by most others. In that case, metric 1 will indicate marginalization, while metric 2 will not give this signal.
Even though we performed data collection on the individual parameters, we scaled the data to the group level so that we can have conclusions for the social dynamics. For this reason, we calculated the percentage of marginalized agents at the group level for each simulation separately based either on metric 1 or 2, these variables were named "Students_ marginalized" and "Students_ marginalized _attraction", respectively.

Table 3: Categorization of Simulations according to the values of Max_jud and Stdev_char.

Simulation type	**Variable value**
Low Tolerance	Max_judg = 0.1
Medium Tolerance	Max_judg = 0.2 or 0.3
High Tolerance	Max_judg = 0.4 or 0.5
Low Diversity	Stdev_char = 0.1
Medium Diversity	Stdev_char = 0.2 or 0.3
High Diversity	Stdev_char = 0.4 or 0.5

4 FINDINGS: THE SOCIAL PHYSICS OF BULLYING

We borrow the term "social physics" from (Pentland 2014) to indicate that we study the dynamics and mechanisms by which ideas, behaviors, and interactions emerge and spread within human populations. The initial simulation experiments with MARG shed light on our research question. Pearson correlations between the group exclusion based on the two different metrics ("Students_ marginalized" and "Students_ marginalized_attraction") showed that these variables were highly correlated: r=0.94 $P \leq 0.001$ for Scenario 1 and r=0.9 $P \leq 0.001$ for Scenario 2. Due to the high correlation, we decided to proceed with the presentation of results for "Students_ marginalized", which is based on Metric 1. For Metric 1, we calculated the descriptive statistics for all simulations of Scenario 1 (see Table 4) and 2 (results in the text). Due to the high variability of the results in Scenario 1, we performed a linear model analysis (see Table 5) and mixed model analysis. The linear model displayed a better explanation for the variance of the results.

In scenario 2, where the idealized values are all set to 1, the values of the agents' characteristics (internal and external) will seldom coincide because agents' characteristics have a mean value of 0.5. Hence, during the evaluation of interactions, most agents will have a negative experience with all others resulting in a high exclusion index. Therefore, agents are driven towards marginalization. Nevertheless, when tolerance is high, the percentage of marginalized students decreases because high tolerance broadens the agents' preferences, and thus, they find other partners attractive even though they have characteristic values far from the 'ideal' ones. This percentage of marginalized students is even lower when combined with higher diversity in the values of agents' characteristics because a high diversity in characteristics increases the chance that some agents have values close to the "ideal" ones. The descriptive statistics of Metric 1 for Scenario 2 showed 100% of students marginalized per simulation except for the cases where we have simulations characterized by high tolerance. When tolerance is high, medium, or low combined with high diversity, we have, on average, 83% of students marginalized per simulation, 93% of students marginalized per simulation, and 97% of students marginalized per simulation, respectively.

The results are more diverse when we look at Scenario 1 in which the idealized values coincide with the average value of the agents' characteristics (Table 4). As a result, the more a student's characteristics match the average values, the more likely it is for this agent to be included and vice versa. As in scenario 2, high tolerance reduces group marginalization because it broadens individual preferences. However, in contrast to Scenario 2, diversity has the opposite effect. The combination that favors marginalization is high diversity and low tolerance (100% of students marginalized per simulation) and medium diversity and low tolerance (98% of students marginalized per simulation) (Table 4).

On the other hand, low diversity favors low rates of exclusion (high rates of inclusion). Low diversity and high tolerance yield an average of 10% of students marginalized per simulation and low diversity and medium tolerance an average of 20% of students marginalized per simulation (Table 4). Interestingly, even though we see conditions leading to 100% of students marginalized per simulation, we do not see the opposite, which is simulations with 0% marginalization. The lowest average marginalization was 10% under the conditions of high Tolerance and low Diversity. It seems, thus, that in our model, the marginalization is more easily achieved than inclusion.

Table 4: The table shows the average percentage of marginalized students (based on Metric 1) per simulation for all simulations under specific conditions of tolerance and diversity. The results are for the simulation of Scenario 1 (total of 3125 simulations). The parentheses show the 1st and 3rd quartile for the specified result.

	Low Tolerance	**Medium Tolerance**	**High Tolerance**
Low Diversity	49% (40%-55%)	20% (14%-26%)	10% (5%-15%)
Medium Diversity	98% (100%-100%)	50% (35%-62%)	26% (20%-34%)
High Diversity	100% (100%-100%)	87% (75%-100%)	50% (40%-60%)

To better understand the effect of the input parameters (i.e. stdev_char, the max_judg, the attraction_change, and the num-students, Table 1) on the percentage of students marginalized per simulation (Metric 1), we performed a linear model analysis. We first ensured that parameters were not collinear and that the assumptions of linearity, homoscedastic, and normality of residuals are met. Results of the linear model analysis (using the lm function in R) are shown in table 5. The model explains a significant amount of the variance (Multiple R^2: 0.83, Adjusted R^2: 0.83). The input variables, Stdev_char (the diversity of agents' characteristics) and Max_judg (tolerance) parameters had a significant ($p < 0.001$), opposite but similar effect in size. None of the other input parameters had a significant effect on the percentage of marginalized students.

Max_judg and stdev_char have antagonistic effects: as the diversity among the values of the agents' characteristic increases, the percentage of students marginalized per simulation increases, and as the value of tolerance increases, the percentage of students marginalized per simulation decreases (Table 5). The effect is also visible in Table 4, where the diagonal combinations (low diversity-low tolerance, medium diversity – medium tolerance, high diversity – high tolerance) display similar simulation outcomes on average. Diversity increases marginalization because of the higher the diversity, the higher the deviation of the agents' characteristics from 0.5 (the idealized value in scenario 1). Thus, the more likely that agents evaluate interactions as negative. Note that this result is in contrast with scenario 2, where idealized values are set to 1. In scenario 2, a higher diversity decreases exclusion incidents since the higher the diversity, the higher the chance that agents' characteristics reach the idealized value of 1.

Table 5: Results of the Linear Model for Metric 1 and Scenario 1.

Linear Model			
Metric	**Effect**	**Standard error**	**P-value**
Intercept	0.58	0.01	<0.001
Stdev_char	1.41	0.02	<0.001
Max_judg	-1.53	0.02	<0.001
Attraction_change	-0.02	0.08	0.84
Num_students	<0.01	<0.01	0.69
Adjusted R^2: 0.83			

The limitations of the model include:

- The evaluation of each characteristic is polarized to either positive or negative, which leads to more extreme total evaluations. In addition, all characteristics are valued the same, whereas, in reality, some characteristics may be more important than others.
- We assumed ten external and ten internal characteristics per agent, and the effect of a different number of characteristics was not explored.
- All students share the same ideal values: a fact which implies the effect of one culture. Moreover, we have set the same ideal cultural value for all internal characteristics and the same ideal cultural value for all external characteristics.

In future work, we plan to look further into the aggregation of individual marginalization and the relationship of individual characteristics to the overall marginalization. Finally, we want to perform an extensive network analysis to track group formations among the students.

5 DISCUSSION: THE SOCIAL ETHICS OF BULLYING

With MARG, we were able to observe the emergence of intensified marginalization (defined with the use of thresholds and the metrics) for multiple parameter setups. The emergence occurred from everyday interactions at the university using one process from the context of inclusion-exclusion, the process of selecting interaction partners. Since we did not include more processes, we can argue that MARG simulates the societal stress of marginalization in students and not the actually manifested marginalization. The relation of the simulated vs. actual exclusion events cannot be estimated without further analysis. Nonetheless, the results raise questions about the "social ethics" of bullying. The latter term is highly contentious, as are the terms commonly related to it, such as exclusion and marginalization. However, generally speaking, most people find something "immoral" or at least morally problematic in the behaviors to which such terms commonly refer. On the other hand, most people do not find it morally problematic that individuals (e.g., university students) like to hang out with people whom they admire. MARG simulations showed that the latter could all too easily lead to the former.

Our modeling efforts here are part of a growing movement in the modeling & simulation profession toward surfacing the ethical assumptions and implications in model formulation, construction, and execution (Shults and Wildman 2019; Shults, Wildman, and Dignum 2018). In this context, we want to highlight the need to focus on *social* ethics as part of this ongoing conversation. All too often, ethical debates attend only or at least primarily to *individual* ethics; e.g., what rule should an individual follow, or what goals should the individual set? However, as models like MARG demonstrate, the social physics of human life is non-linear, and the emergent effects of goal setting are often far from what individuals intend. By constructing models that are attentive to social-ethical concerns in collaboration with philosophical ethicists and other stakeholders, computer scientists can contribute new tools and insights to conversations about how to mitigate the deleterious effects of social marginalization.

Within the boundaries of our model's limitations, we can draw conclusions on the social physics of bullying. Our findings are similar to those of the Schelling model that demonstrated how small shifts in preferring similar neighbors quickly lead to increased segregation between simulated agents (Schelling 1971). Unlike the Schelling model, however, MARG provides more insights about the drivers of the issue. The most notable conclusions from our simulations were the positive effect of tolerance on inclusion and the mixed effect of diversity on marginalization. The former is easily explained by the fact that tolerance determines individual preference, and a broad individual preference will result in a positive evaluation of the interaction partner and enhance inclusion. Explaining the latter requires more reflection. We would expect diversity to enhance the chance of students being included. Since we are simulating a monoculture with stabilized norms,

diversity has a positive impact on inclusion when the culture idealizes characteristics outside of the average range. This is because, in our context, diversity introduces agents with characteristics closer to the ideal values. On the other hand, diversity worsens marginalization when a culture idealizes the average student because it increases the chance of deviation from the ideal value.
Based on our results, teaching tolerance is a universal strategy to combat bullying. Unfortunately, integrating tolerance into a person's thinking takes time, and therefore it can be considered a long-term strategy. On the other hand, creating groups with a preferred level of diversity is a short-term strategy, but it holds a risk. To decide about the diversity of a group requires a deep understanding of how ideal values relate to average characteristics in the group and knowledge of student characteristics. As it can be very difficult and ethically problematic to assess individuals' characteristics and store data about them, this short-term strategy applies only to specific groups such as small classes with consistent attendance. Overall, apart from focusing on altering the individuals' level of tolerance, educators and other stakeholders might also pay more attention to the function played by excessively high micro-level ideals in generating macro-level marginalization.

ACKNOWLEDGMENTS

The authors are grateful to the University of Agder for a grant to support the "Computer Tools for Modeling Social Conflict" project, which provided research funding for Themis Dimitra Xanthopoulou and Ivan Puga-Gonzalez.

REFERENCES

Blood, G. W., and I. M. Blood. 2016. "Long-term Consequences of Childhood Bullying in Adults who Stutter: Social Anxiety, Fear of Negative Evaluation, Self-Esteem, and Satisfaction with Life." *Journal of Fluency Disorders* vol. 50:pp. 72-84.

Cohen, J. W., and R. A. Brooks. 2018. "The Construction of School Bullying as a Social Problem." In *The Cambridge Handbook of Social Problems*, edited by A. J. Treviño, pp. 237-252. Cambridge: Cambridge University Press.

Cornell, D., A. Gregory, F. Huang, and X. Fan. 2013. "Perceived Prevalence of Teasing and Bullying Predicts High School Dropout Rates." *Journal of Educational Psychology* vol. 105 (1):pp. 138-149.

Espelage, D. L., and S. M. Swearer. 2010. "A Social-ecological Model for Bullying Prevention and Intervention: Understanding the Impact of Adults in the Social Ecology of Youngsters." In *Handbook of Bullying in Schools: An International Perspective.*, edited by S. R. Jimerson, S. M. Swearer and D. L. Espelage, pp. 61-72. New York, NY, US: Routledge/Taylor & Francis Group.

Hofstede, Geert H., Gert Jan Hofstede, and Michael Minkov. 2010. *Cultures and organizations: software of the mind : international cooperation and its importance for survival*. Rev. and expanded 3rd ed. ed: United States: McGraw-Hill USA.

Lund, I. 2017. Mobbing i høyere utdanning, fleip eller fakta. https://www.universell.no/.

Maeda, Yoshinobu, Kazuya Anezaki, and Yumi Takeuchi. 2006. "An agentbased model for simulating the group dynamics involved in excluding a minority." Proc. 1st World Congress on Social Simulation.

Mathieson, J., J. Popay, E. Enoch, S. Escorel, M. Hernandez, H. Johnston, and L. Rispel. 2008. "Social Exclusion Meaning, Measurement and Experience and Links to Health Inequalities." *A review of Literature. WHO Social Exclusion Knowledge Network Background Paper* vol. 1:pp. 91.

Paluck, E. L., H. Shepherd, and P. M. Aronow. 2016. "Changing Climates of Conflict: A Social Network Experiment in 56 Schools." *Proceedings of the National Academy of Sciences* vol. 113 (3):pp. 566-571. doi: 10.1073/pnas.1514483113.

Peace, R. 2001. "Social Exclusion: A Concept in Need of Definition?" *Social Policy Journal of New Zealand* (16):pp. 17-36.

Pentland, A. 2014. *Social Physics : How Good Ideas Spread the Lessons from a New Science*. New York: Penguin.

Salmivalli, C., K. Lagerspetz, K. Björkqvist, K. Österman, and A. Kaukiainen. 1996. "Bullying as a Group Process: Participant Roles and their Relations to Social Status within the Group." *Aggressive Behavior* vol. 22 (1):pp. 1-15.

Schelling, T. C. 1971. "Dynamic Models of Segregation." *Journal of Mathematical Sociology* vol. 1 (2):pp. 143-186.

Schott, R. M., and D. M. Søndergaard, eds. 2014. *School Bullying: New Theories in Context*. Cambridge: Cambridge University Press.

Shults, F. L., and W. J. Wildman. 2019. "Ethics, Computer Simulation, and the Future of Humanity." In *Human Simulation: Perspectives, Insights, and Applications*, edited by Diallo S. Y., Wildman W. J., Shults F. L. and Tolk A., pp. 21-40. Springer.

Shults, F. L., W. J. Wildman, and V. Dignum. 2018. "The Ethics of Computer Modeling and Simulation." 2018 Winter Simulation Conference (WSC), Gothenburg, Sweden.

Thawiworadilok, Chaiwat, Mohsen Jafari Songhori, and Takao Terano. 2017. "Coping with Bullying in the Classroom Through Agent-Based Modeling." Agent-Based Approaches in Economics and Social Complex Systems IX, Singapore, 2017//.

Thibaut, John W., and Harold H. Kelley. 1959. *The social psychology of groups.* New York: John Wiley & Sons, INC.

Tseng, Shih-Hsien, Chien-Kuo Chen, Jia-Chen Yu, and Ying-Chi Wang. 2014. "Applying the agent-based social impact theory model to the bullying phenomenon in K–12 classrooms." *SIMULATION* 90 (4):425-437. doi: 10.1177/0037549714524452.

Turner, H. A., D. Finkelhor, A. Shattuck, S. Hamby, and K. Mitchell. 2015. "Beyond Bullying: Aggravating Elements of Peer Victimization Episodes." *School Psychology Quarterly* vol. 30 (3):pp. 366-384. doi: 10.1037/spq0000058.

Wolke, D., and S. T. Lereya. 2015. "Long-term Effects of Bullying." *Archives of Disease in Childhood* vol. 100 (9):pp. 879-885.

AUTHOR BIOGRAPHIES

THEMIS XANTHOPOULOU is a doctoral candidate at the University of Agder currently working on computer tools to address bullying. She is passionate about interdisciplinary research and draws knowledge from diverse scientific fields. Her email address is themis.d.xanthopoulou@uia.no.

IVAN PUGA-GONZALEZ is a post-doctoral research fellow at the University of Agder. His research interest is on the self-organization of complex social systems. More specifically, on how complexity at the group level may emerge from local interactions and simple heuristics at the individual level. He is particularly interested in human and non-human primates' social systems. He holds a Ph.D. in Theoretical Biology from the University of Groningen. His email address is ivan.puga-gonzalez@uia.no.

F. LERON SHULTS is Professor at the Institute for Global Development and Planning at the University of Agder in Kristiansand, Norway, and Scientific Director of the Center for Modeling Social Systems at NORCE. He has published 18 books and over 120 scientific articles on topics within a variety of fields, including social science, philosophy, cognitive science of religion, and computer science. His email address is leron.shults@uia.no.

ANDREAS PRINZ is Professor for systems development at the department of information and communication technology at the University of Agder, Campus Grimstad. He is interested in systems modeling and formal languages to express systems. He has experience in modeling systems, both technical and social, and in validating and verifying system models. His email address is andreas.prinz@uia.no.

Paper C

The Problem with Bullying: Lessons Learned from Modelling Marginalization with Diverse Stakeholders

Themis Dimitra Xanthopoulou, Andreas Prinz, F. LeRon Shults

Themis Dimitra Xanthopoulou, Andreas Prinz, and F. LeRon Shults. The problem with bullying: Lessons learned from modelling marginalization with diverse stakeholders. In Marcin Czupryna and Bogumil Kamiński, editors, Advances in Social Simulation, Springer Proceedings in Complexity,
pages 289–300. 2022, Springer Nature

The Problem with Bullying: Lessons Learned from Modelling Marginalization with Diverse Stakeholders

Themis Dimitra Xanthopoulou, Andreas Prinz, and F. LeRon Shults

Abstract While building a simulation model to gain insights on bullying interventions, we encountered challenging issues that forced us to reconsider the concepts we model. We learned lessons about the need for quality assurance and a more demanding construction process when building models that aim to support decision making. One of the lessons is that even academically accepted concepts such as "bullying" can be ambiguous. Experts and interested parties do not agree about how to define and use the term bullying. Indeed, before we can model "bullying", we need a shared understanding, of its meaning. Otherwise, insights from the model could be misinterpreted and lead to misleading conclusions. Concepts are inherently imprecise and contain grey areas. Although this may be true, not all of them are ambiguous. For the scope of this paper, ambiguity implies that the same word is used to point to different concepts. For different reasons, bullying has evolved to point to different concepts for different people and sometimes even for the same person. We propose to solve these challenges by identifying which concrete bullying behaviors to target, and by focusing on simulation models for interventions addressing those behaviors.

Keywords Bullying · Social simulation · Formalization · Interventions · Agent-based modeling

1 Introduction

There are several reasons why social simulation appears to be a promising tool for research on (and the facilitation of) conflict resolution. First, the formalization of theories and causal claims about a conflict within a computational model themselves help to clarify the tangible issues surrounding the conflict and to foster dialogue about possible ways of resolving it [1]. Moreover, a single computational architecture

T. D. Xanthopoulou ( ) · A. Prinz
University of Agder, Jon Lilletuns vei 9, 4879 Grimstad, Norway
e-mail: themisdxanthopoulou@yahoo.com

F. L. Shults
University of Agder, Universitetsveien 19, 4630 Kristiansand, Norway

M. Czupryna and B. Kamiński (eds.), *Advances in Social Simulation*,
Springer Proceedings in Complexity, https://doi.org/10.1007/978-3-030-92843-8_22

for an "artificial society" can integrate multiple disciplinary perspectives, which is crucial when dealing with complex interpersonal or inter-group conflicts [2]. Finally, insights from simulation experiments can inform debates among stakeholders and policy-relevant decisions in the real world [3].

At the beginning of this project, we viewed bullying as a complex type of social conflict that would benefit from a social simulation approach. A simulation model of bullying would improve existing intervention programs and proposed solutions, which so far have had mixed results [4–6]. The topic of bullying has been researched by multiple disciplines such as criminology, psychology, sociology etc. In a telling comment about the state of bullying research, one of the keynote speakers at the Anti-Bullying Forum expressed the opinion that there is no theory of bullying. We believed that a social simulation of "bullying" could help to integrate the different points of view, enable intervention testing, provide reasons and solutions for inconsistent outcomes, and contribute to a robust theory of bullying.

Based on these premises, we set out to create a model of bullying. Our model did turn out to be useful, but our efforts with various stakeholders to improve the model led to some important lessons. In Sect. 2, we describe our methodology to create a bullying model. We present the results of our efforts in Sect. 3 and discuss them in Sect. 4. From there, we propose a solution in Sect. 4.3 and conclude in Sect. 5.

2 Methodology

The goal of the planned methodology was to achieve the construction of a simulation model of bullying for interventions. More specifically, the aim of the model was to understand the emergence of bullying and to test which interventions would be successful in preventing this emergence. The purpose can be classified as explanatory [7] insofar as we were trying to figure out both the causal architectures of bullying and to determine the reasons behind mixed outcomes in established intervention programs. However, we can also describe the model's purpose as exploratory in the sense that it attempts to provide a deeper understanding of the target system [8] and a deeper exposition of theories [7] around the concept of bullying.

Our plan was to first capture the important dynamics and then add components to the model so that we can map the intervention mechanisms. To begin with, the construction of a social model is typically the work of an interdisciplinary team [9]. A social model of bullying needs a bigger network of collaborators because it requires more perspectives due to the nature of the subject. Moreover, bullying behaviors are considered a complex issue [10]. The construction of complex models is comprised of multiple steps. With each step, the model is extended and becomes more and more complex. Finally, the explanatory character of the model alongside the prospect of supporting decision making means that our model needs to meet better quality standards. To meet the quality requirements, to address the complexity, and to account for the lack of bullying expertise in our team, we designed a methodology that drew on the principles of the agile programming approach: “iteration” and “flexibility” [11]. We planned the following steps:

The first step was to conduct initial literature search, operationalise bullying, and select the model focus. The second step was the construction of a working simulation model of bullying to act as a starting point, a minimum viable product (MVP) to initiate the feedback sessions. The third step is the presentation of the first version to subject matter experts for feedback. Next, we planned to correct and expand the first model. The fifth step was the presentation of the second model version to subject matter experts for feedback and request data for validation (where possible). The process was to be repeated until a satisfactory level of agreement and model capacity was reached. The final step was the validation of the model using interviews. The goal was to use the model to understand the impact of interventions. All in all, our planned input was: literature search, feedback sessions with experts, and interviews. The input methods were to be used as supplements, where needed, throughout the model construction process.

3 Results

In this section, we summarize the results of the model construction process. First, we explain the modelling choices for the first version of the bullying model. Then, we present the feedback we received from our first session with stakeholders. Finally, we display our findings from the intensive literature search and interviews regarding the bullying concept.

3.1 First Modelling Choices

We chose to focus on university bullying due to our context. Currently, there are two dominant definitions of bullying corresponding to different paradigms [12]. The oldest one, introduced by Olweus, which we will call the “Olweus definition” defines bullying as the “negative actions” of one or multiple persons towards one person, “repeated” over time, when the actor/s and the receptor of the behavior have “asymmetric power relationship” [13]. To build our first model version, we selected a recently evolved definition, which we will call “Schott and Søndergaard definition”, that views bullying as “... an intensification of the processes of marginalisation that occur in the context of the dynamics of inclusion/exclusion, which shape groups. Bullying happens when physical, social, or symbolic exclusion becomes extreme, regardless of whether such exclusion is experienced and/or intended.” [12]. The second definition reflects a new paradigm on viewing bullying that puts more focus on the social dynamics component. This choice was the outcome of discussions with our main bullying subject matter expert.

We decided we would use the agent-based approach for our simulation model as it allows the observation of emergent phenomena. This approach organises the subject matter knowledge around three main components: agents, attributes (environment, agent, general), and behaviors (including events). Such an approach is very close to the modes of thought of interventionists and adapts well to the needs of the jurisdictional system. Drawn from the selected bullying definition, the behavior we chose to model is the exclusion of university students from dyadic interactions, and negative experiences in dyadic interactions during leisure time at the university. We considered a student to be bullied when the percentage of negative interactions and exclusion experiences to overall interactions is high.

3.2 Diverse Feedback

To invite feedback from subject matter experts, we presented our work at the Anti-Bullying Forum [14]. We enquired whether our first version seemed intuitive and what additions we would need to make to proceed with a more intuitive model. The model presentation was in the form of a poster with a simplified explanation of how the model works as well as discussions stemming from the poster presentation. Most of the participants at the Anti-Bullying Forum were not familiar with computer modelling.

Bullying experts and practitioners reserved a neutral attitude towards the first model version during the feedback sessions. They did not seem triggered negatively by our model with its agent characteristics and rules, but they did not endorse it either. An exception was our selected definition, which raised some questions due to the conflict among the different paradigms. Our primary goal during our interactions with conference participants was to invite their suggestions for improvement based on their individual understandings and research agenda.

The input we received was very diverse, a fact that might not be so surprising since bullying experts come from such diverse disciplinary backgrounds. Each recommendation translated into one or more model variables, and one or more agent behaviors. Our initial model included 23 parameters, some of which are mathematical with the potential to contain up to 100 elements. Table 1 presents the input we received divided in 5 categories.

3.3 Reflection on the Modelling Process and Further Literature Research

The next step after the Anti-Bullying Forum was the correction and extension of the bullying model, which proved a challenging task. Gaining a distance from the modelling process and moving to a period of reflection enabled us to acknowledge this

Table 1 Input grouped in categories

Personality	Psychological needs	Personal skills	Context	Social
Personal tendency to isolate	General acceptance	Self-efficacy	Friendship networks	Social capital
Resistance to belonging	Acceptance from friends	Sociometric status and perceived popularity	Teacher as agents	Social impact
Aggressiveness	Belonging	Numb blindness		Social influence
		Connection and disconnection from self and others		Peer influence
		Questioning personal perception		Social norms
		Social and emotional learning		Effect of role models
		Filtering information		Effect of leaders
		Goal setting		

difficulty and to distinguish it from the usual challenges of the modelling process. Our challenge was to select the aspect which would be added to the next model version. The diversity of the feedback shows that bullying researchers hold different views on what is considered crucial in determining bullying behaviors. This realisation is intensified by the fact that we received feedback from a limited amount of people. Upon attendance of different sessions at the Anti-Bullying Forum , we discovered even more perspectives that might be included in our model. To make the most of the reflection process, we decided to supplement our input with additional literature searches. The next subsections present our findings.

Different Understandings of Bullying Except for the different academic definitions and thus understandings of what bullying is [12], we found out that perceptions of bullying differ between academics and non-academics. In some studies [10, 15], researchers have noticed that views of what constitutes bullying did not coincide with the dominant definition at the time (the Olweus definition). To be more specific, in the study with teachers [10], teachers did not consider parts of the definitions important to classify a behavior as bullying and one teacher changed what she considered bullying after hearing the definition from the researchers. In addition, in the study mentioned in [15], students changed their answers in a bullying survey after being given the definition. Interestingly enough, teachers did not judge the characteristics of the behavior itself to evaluate whether an observed behavior is bullying, but also factors such as a student's fitness to be called a victim, their judgement on whether the student "deserved" the behavior, the "normalcy" of the behavior, and the student perception (as suggested by the second paradigm) [10]. The study mentioned in [16] also showed that the emotional effect of the behavior on the student at the receiving end of it counted as a factor for whether other students classified the behavior as bullying.

Measuring Bullying One way to assess bullying is by using surveys. Cornell and Bandyopadhyay [17] point out that some surveys employ definitions to clarify what they mean by bullying while others use simpler versions of one of the definitions, which include ambiguous elements. Furthermore, the Juvenile Victimization Questionnaire avoided definitions altogether and asked 2 questions for the categories "bullying" and "teasing and emotional bullying". The categories were specified with the behaviors "chasing, grabbing hair or clothes, making you do something you did not want to" for the bullying category, and "feeling bad or scared because of calling names, saying mean things, or saying they did not want you around" for teasing or emotional bullying. The Olweus Bully/Victim Questionnaire (referring to the cited version [18]) uses the combination of definitions and behavior lists to achieve concept clarification. It includes more behavior categories than the Juvenile Victimization Questionnaire, such as intentional exclusion from a group of friends. In addition, they present more behavior examples including the general description of "other hurtful things like that", which is open to interpretation. As the questionnaire proceeds, it introduces the Olweus definition to restrict what counts as bullying and adds a note that says not to include playful teasing and behaviors that are not repetitive or behaviors between individuals without power imbalance. Apart from surveys that ask people whether they have been bullied, there are surveys that ask others to nominate who has been bullied [17]. These surveys seem to operate under the same methods of concept clarification.

Another method to measure bullying instances are naturalistic observations. In one example, observers counted as bullying the "aggressive events" in which there is a power relationship between the aggressor and the receiver of the aggressive behavior [19]. Finally, apart from the survey-based-interviews or interviews that followed surveys [17], researchers have utilized interviews to explore bullying. It is not clear how interviewers measure bullying. In study [20], the authors mention that they based their assessment on the description of experiences but do not refer to whether they exposed interviewees to their perceptions of bullying definitions or examples of bullying behaviors. Similarly, in study [21], the authors mention the fact that they asked parents and children whether their children or they themselves had been bullied but the authors do not explicitly state whether they provided definitions or examples of bullying behaviors to interviewees to understand their perception of bullying.

Evolution of the Bullying Concept It turns out that "bullying" evolved into an umbrella concept that accommodates various and quite diverse behaviors [22].

According to Schott and Søndergaard [12], the concept history traces to the term "mobbing", understood as the attack on one person by a group of people. Later, with the help of the media, the term bullying started to convey behaviors with varying intensity and effect. Cohen et al. give the examples of non-physical behaviors such as social exclusion, criminal behaviors such as predatory sex crimes, mutual teasing, and rough-housing to account behaviors that were given the label of bullying. At the same time, they mention that the concepts of "bully" and "victim" changed in such a way that most children can be categorized as either the former or the later [22]. Possibly connected to the evolving character of the bullying concept, researchers have discovered perceptual differences between different stakeholders and researchers when it comes to what is categorized as bullying [10, 16, 23, 24].

Informal Interviews We conducted interviews to further investigate the concept of bullying. We discussed bullying with several colleagues to test the variance even within one institution, namely, our university. The University of Agder has established a report system that is visible immediately when one visits the main university website [25]. The link for the report system contains information about how to use the system alongside information about bullying. However, it does not list the behaviors that fall into the category of "bullying".

We chose to interview colleagues from different departments, in different positions (organisational or research related), and in various levels of decision making regarding bullying issues. We asked them to describe what bullying constitutes for them, to point to specific behaviors and whether the behavior we had included in our model was registered under the bullying concept from their perspective. Most of our interviewees faced difficulty when trying to identify bullying behaviors. Interestingly enough, considering the small number of interviewees, they did not agree on whether the behavior we modelled in our model was a bullying behavior. The hypothesis that there is a shared understanding of the bullying concept at our university was falsified.

4 Discussion

From Sect. 3.3, we can extract the following:

- Academics and non academics do not agree oh what is important to define bullying. In general, people have diverse views on what criteria to use to define a behavior as bullying.
- The categorization of bullying in practice does not involve only the assessment of the action itself but subjective factors such as the effect on the "victim", how much the observant likes the "victim" etc. This might explain why one observed behavior might be interpreted differently by different observants.
- Bullying is evolving to include more and more behaviors. Nevertheless, we cannot be certain that everyone has the same access to the new aspects of the concept.

The disagreement on characterization criteria, the subjectiveness in evaluation, the inclusion of more and more behaviors, and the different access to the concept evolution make the concept of bullying unmanageable. In this section, we start by explaining the issues behind modelling "bullying" based on our findings. We then continue with the evaluation of the different ways we talk about bullying, characterized by different abstraction levels in our mission for a modelling alternative.

4.1 Expectations Behind Modelled Concepts

Before we go into the analysis we should refer to concept fuzziness and ambiguity. Concept imprecision or fuzziness implies that there are grey zones in concepts. When encountered with a grey zone, we are not sure whether the concept can be applied to describe our observation. A "fuzzy" concept is something that cannot be avoided. Concerns over concept fuzziness have been addressed before such as in the case of the social model "The Status-Arena" and the concept of "Rough and Tumble" [26]. In essence, modelling helps with concept precision since it exposes aspects of each concept ontology. Term ambiguity implies that a term is used to describe two different concepts. An example is the term "crane", used to describe a type of bird or a machine. Apart from the distinction between fuzzy concepts and ambiguous terms, there is the moral judgement of an observed action. Due to the subjective nature of morality, moral judgements of the same observation vary.

We can hypothesize the following scenario: bullying starts by the meaning of "all against one physical violence". This might have been a manageable use of the concept but then, the media extend the meaning of the term. In search of an explanation for deeply shocking events, such as student suicides and mass shootings [22], journalists tie more behaviors to the term. Researchers contribute to the trend by adding more dimensions to bullying with the development of definitions. Depending on individual media access and other sources, people develop the concept differently. On a collective level, bullying points to different concepts of actions ranging in intensity, context etc. It is very easy to identify "crane" as an ambiguous term since the two concepts involved do not have any similarities. It is much harder to identify bullying as ambiguous since the concepts involved are all interactions of some kind. We believe that a more accurate characterization of bullying is to say that it is a moral judgement. You will rarely hear someone endorsing "bullying" behaviors (maybe only in cases of intended revenge). We propose that the term developed to basically include negatively judged behaviors in interactions among people. This explains why teachers and students evaluate the effect of the behavior on the "victim", and the personality of the "victim" to assess whether a behavior is bullying. It is because these factors affect their moral sensitivities. Such a term is very useful to assist the Anti-Bullying movement. While this is perfectly in line with the progress of a social movement, it is not helpful when it comes to being a modelled concept of actions.

To successfully use an explanatory model of bullying, it is very crucial to agree on what actions constitutes bullying. Without a shared understanding, it is not possible for interested stakeholders, including decision makers, to know how and where model insights apply. However, the state of the term "bullying" and the status of model communication methods cannot promise clarity over what is modelled and what is not. Model communication is a set of techniques, such as model documentation, to help us illustrate what the model does and what it includes. Nevertheless, in practice, even if documentation is available, model specifics are not always understood [27]. Consequently, model documentation is not the best way to clarify our definition of bullying to people without modelling experience, such as bullying experts. They will typically assume that we follow their definition. In the following sections we try to identify an alternative to the use of the term of bullying for modelling purposes.

4.2 Exploration of Alternatives

In Fig. 1, we have mapped different items related to the word bullying and ordered them by their abstraction level in an effort to find an alternative modelling content. Lower level of abstraction means that the item is more closely connected to the object, in this case the observed action.

The first items correspond to the different definitions given by researchers. Different definitions point to different concepts. For example, the behavior we used in our first version model [28], would be categorized as bullying using the Schott and Søndergaard definition but would not under the Olweus. Definitions are still unfit to serve as a modelling content as non academics do not use their elements to categorize behaviors as bullying. Consequently, were we to employ definitions, we would face issues with validating our models as measurements do not typically include observations from academics. In addition, it would still be hard to communicate clearly insights and limitations of the model to interested stakeholders.

Bullying modes are on a similar level of abstraction as definitions. Bullying modes specify behaviors by limiting them in a specific context. For example, physical bullying needs to include physical behaviors. Bullying modes are still ambiguous since they inherit the same properties as the term bullying in a more specified context.

When we move on to the level of behaviors, we notice that it is easier to distinguish among different behaviors. For example, it is easy to say which behavior is a "name-calling" behavior and which is "hitting". Modelling behaviors might imply that we depart from the term bullying since not all behaviors are unanimously categorized as bullying. In addition, as with the Rough and Tumble, we may still face grey areas and confusions over judgement. Nevertheless it is much clearer to distinguish between two types of behaviors and thus for theories and explanations to emerge.

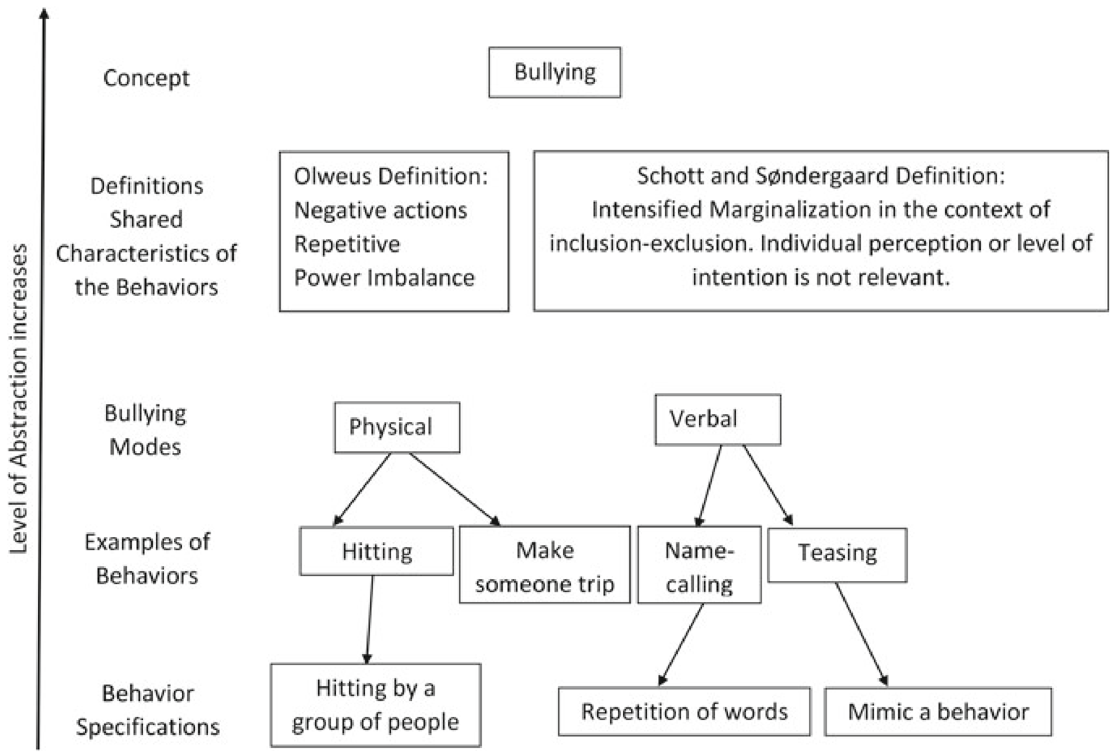


Fig. 1 Bullying at different levels of abstraction

4.3 Proposed Solution

We argue that the solution is to model concrete behaviors. A model of "hitting" or "name-calling" gives less space for speculation and does not introduce uncertainty about the results. Even though there is less confusion of what the model includes, these behaviors are still complex in nature. Some anticipated implications of following the proposed solution are:

– The feedback from literature and experts used in the model construction process will be more straightforward.
– Stakeholders will be forced to clarify what are the issues they need to solve instead of mentioning abstract notions such as "bullying".
– Stakeholders can readily apply insights from models without worries of misinterpretations.
– Multiple models will need to be used in combination to achieve the resolution of a variety of behaviors.
– On the negative side, we expect less people to be interested in these models since bullying has become a catching phrase.

5 Conclusions and Word of Warning

The feedback we received from stakeholders on our explanatory model of bullying led us into a deeper investigation of the concept using literature search and interviews. We found out that bullying is evolving and expanding, matches more than one concepts, and fits better the form of a moral judgement than an action concept. The ambiguity of the term leads to inconsistent measurements. Considering the term "bullying" as a moral judgement might explain why there is a big range of understandings of what bullying is, and why factors such as the relationship between observer and behavior recipient, and the result of the behavior, play a role in characterizing the behavior as "bullying".

Bullying definitions and bullying modes are less abstract ways to talk about the same issues. Nevertheless, even on this level of abstraction, we face similar discrepancies. We propose to model concrete behaviors and to move away from "bullying" so as to avoid misunderstandings regarding model usage. The conclusion is intensified by the low level of model communication. More work needs to be done to identify possible implications of using concepts of concrete behaviors to test for interventions.

Our study concerns bullying, but the same issues appear whenever concepts are not only fuzzy, but ambiguous. The bullying community does not seem to be aware of the issue and the simulation community has not established procedures to assess the fitness of concepts. We encourage modellers to consider whether their modelled concepts might raise similar issues as the ones we faced. In that case, we suggest the further specification of models that target specific behaviors and we encourage modellers to avoid ambiguity that hinders theoretical clarity and successful practical interventions.

References

1. Shults, F.L., Gore, R., Wildman, W.J., Lynch, C., Lane, J.E., Toft, M.: A generative model of the mutual escalation of anxiety between religious groups. J. Artif. Soc. Soc. Simul. **21**(4), 1–25 (2018)
2. Shults, F.L., Wildman, W.J.: Human simulation and sustainability: ontological, epistemological, and ethical reflections. Sustainability **12**(23) (2020)
3. Mehryar, S., Sliuzas, R., Schwarz, N., Sharifi, A., van Maarseveen, M.: From individual fuzzy cognitive maps to agent based models: modeling multi-factorial and multi-stakeholder decision-making for water scarcity. J. Environ. Manag. **250**(109482) (2019)
4. Vreeman, R.C., Carroll, A.E.: A systematic review of school-based interventions to prevent bullying. Arch. Pediatr. Adolesc. Med. **161**(1), 78–88 (2007)
5. Jeong, S., Lee, B.H.: A multilevel examination of peer victimization and bullying preventions in schools. J. Criminol. **2013** (2013)
6. Smith, P.K., Ananiadou, K., Cowie, H.: Interventions to reduce school bullying. Can. J. Psychiatry **48**(9), 591–599 (2003)
7. Edmonds, B., Le Page, C., Bithell, M., Chattoe-Brown, E., Grimm, V., Meyer, R., Montañola Sales, C., Ormerod, P., Root, H., Squazzoni, F.: Different modelling purposes. J. Artif. Soc. Soc. Simul. **22**(3), 6 (2019)

8. Ligmann-Zielinska, A., Siebers, P.O., Magliocca, N., Parker, D.C., Grimm, V., Du, J., Cenek, M., Radchuk, V., Arbab, N.N., Li, S., Berger, U., Paudel, R., Robinson, D.T., Jankowski, P., An, L., Ye, X.: "One Size Does Not Fit All": a roadmap of purpose-driven mixed-method pathways for sensitivity analysis of agent-based models. J. Artif. Soc. Soc. Simul. **23**(1), 6 (2020)
9. Xanthopoulou, T.D., Prinz, A., Shults, F.L.: Generating executable code from high-level social or socio-ecological model descriptions. In: i Casas, P.F., Sancho, M.R., Sherratt, E. (eds.) System Analysis and Modeling. Languages, Methods, and Tools for Industry 4.0, vol. 11753, pp. 150–162. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-30690-8_9
10. Mishna, F., Scarcello, I., Pepler, D., Wiener, J.: Teachers' understanding of bullying. Can. J. Educ./Rev. Can. l'éducation **28**(4), 718–738 (2005)
11. Moyo, D., Ally, A.K., Brennan, A., Norman, P., Purshouse, R.C., Strong, M.: Agile development of an attitude-behaviour driven simulation of alcohol consumption dynamics. J. Artif. Soc. Soc. Simul. **18**(3), 10 (2015)
12. Schott, R.M., Søndergaard, D.M. (eds.): School Bullying: New Theories in Context. Cambridge University Press, Cambridge (2014)
13. Olweus, D.: Aggressive Behavior: Current Perspectives. Springer, Boston (1994)
14. World anti-bullying forum (2019). https://worldantibullyingforum.com/previous-forums/wabf-2019/. Accessed 21 May 2021
15. Vaillancourt, T., McDougall, P., Hymel, S., Krygsman, A., Miller, J., Stiver, K., Davis, C.: Bullying: are researchers and children/youth talking about the same thing? Int. J. Behav. Dev. **32**(6), 486–495 (2008)
16. Guerin, S., Hennessy, E.: Pupils' definitions of bullying. Eur. J. Psychol. Educ. **17**(3), 249–261 (2002)
17. Cornell, D.G., Bandyopadhyay, S.: Handbook of Bullying in Schools: An International Perspective, 1st edn, p. 12. Routledge (2009)
18. Olweus, D.: The Olweus bully/victim questionnaire. https://www.researchgate.net/publication/247979482. Accessed May 2021
19. Hawkins, D.L., Pepler, D.J., Craig, W.M.: Naturalistic observations of peer interventions in bullying. Soc. Dev. **10**(4), 512–527 (2001)
20. Lund, I., Ertesvåg, S., Roland, E.: Listening to shy voices: shy adolescents' experiences with being bullied at school. J. Child Adolesc. Trauma **3**(3), 205–223 (2010)
21. Christensen, L.L., Fraynt, R.J., Neece, C.L., Baker, B.L.: Bullying adolescents with intellectual disability. J. Ment. Health Res. Intellect. Disabil. **5**(1), 49–65 (2012)
22. Cohen, J.W., Brooks, R.A.: The Cambridge Handbook of Social Problems. Cambridge University Press, Cambridge (2018)
23. Boulton, M.J., Trueman, M., Flemington, I.: Associations between secondary school pupils' definitions of bullying, attitudes towards bullying, and tendencies to engage in bullying: Age and sex differences. Educ. Stud. **28**(4), 353–370 (2002)
24. Naylor, P., Cowie, H., Cossin, F., de Bettencourt, R., Lemme, F.: Teachers' and pupils' definitions of bullying. Br. J. Educ. Psychol. **76**(3), 553–576 (2006)
25. Speak up. https://www.uia.no/en/about-uia/speak-up. Accessed 23 May 2021
26. Hofstede, G.J., Student, J., Kramer, M.R.: The status-power arena: a comprehensive agent-based model of social status dynamics and gender in groups of children. AI Soc. (2018)
27. Kolkman, D.: The (in)credibility of algorithmic models to non-experts. Inf. Commun. Soc. 1–17 (2020)
28. Xanthopoulou, T.D., Puga-Gonzalez, I., Shults, F.L., Prinz, A.: Modeling marginalization: emergence, social physics, and social ethics of bullying. In: 2020 Spring Simulation Conference (SpringSim), pp. 1–12. IEEE (2020)

Paper D

Informal Model vs Formal Model: Human Experience vs Conceptualized Model Descriptions

Themis Dimitra Xanthopoulou, Andreas Prinz, F. LeRon Shults

The presented version is the author's submitted version using the dissertation template. This paper was submitted as:

Themis D. Xanthopoulou., Andreas Prinz (2022). Informal Model vs Formal Model: Human Experience vs Conceptualized Model Descriptions [Submitted Paper]

Paper E

Formal ODD: Exposing Model Physics and Model Context

Themis Dimitra Xanthopoulou, Andreas Prinz, Ivan Punga-Gonzalez, F. LeRon Shults

The presented version is the author's submitted version using the dissertation template. This paper was submitted as:

Themis D. Xanthopoulou, Andreas Prinz, Ivan Puga-Gonzalez, F. LeRon Shults (2022). Formal ODD: Exposing Model Physics and Model Context [Submitted Paper]

Poster F

Social Simulation of University Bullying

Themis Dimitra Xanthopoulou, Andreas Prinz, F. LeRon Shults, Ivan Puga-Gonzalez, and Ingrid Lund

This poster has been presented as:

Themis D. Xanthopoulou, Andreas Prinz, F. LeRon Shults, Ivan Puga-Gonzalez, and Ingrid Lund. Social Simulation of University Bullying. 2019. Poster presented at the Antibullying Conference 2019, Dublin, Ireland

World Anti-Bullying Forum
DUBLIN
JUNE 4 - 6, 2019

Social Simulation of University Bullying

UiA University of Agder

Themis-Dimitra Xanthopoulou[1] , Andreas Prinz[1], F. LeRon Shults[1], Ivan Puga-Gonzalez[1], Ingrid Lund[1]

[1]University of Agder, Kristiansand, Norway

Approach

Bullying definition:

*"**Intensification of marginalisation processes** that occur in the context of dynamics of **inclusion/exclusion**, which shape groups. Bullying happens when physical, social or symbolic exclusion becomes extreme, regardless of whether such exclusion is experienced and/or intended."* (Schott and Søndergaard)

SIMULATION OF EXCLUSION-INCLUSION
INTERMEDIATE STEPS : ADDITIONS
FINAL STEP : SIMULATION OF BULLYING

Theoretical Background of this Model

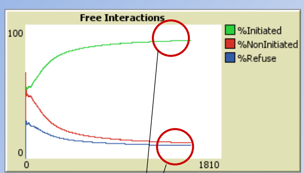
Social Exchange Theory (introduced by Thibault and Kelly, 1959): Interactions have costs and rewards. Good compatibility (in our model evaluation) marks rewards and bad compatibility marks costs.

Culture, Norms and Judgements: Different cultures favor different types of behaviors (Hofstede, Hofstede, & Minkov, 2010). Some people comform to the norms set by society while others do not (Sanaria, 2004). In our model, each individual adjusts an Idealised Cultural Value with a Judgement Variation to make Individual Preferences.

Research Question

"Can intensified marginalisation emerge from selecting interaction partners based on individual preferences informed by cultural norms?"

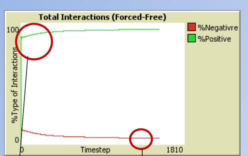
There is equilibrium in interactions after some time..



…depending on the how quickly attractions link change after interaction, Attitude, Learning and Number of Intrinsic Characteristics

In real life this equilibrium may not be reached!

Positive interactions happen when..



….the characteristics of students overlap with the Individual Preferences.

Negative interactions in all other other cases:

- Individual preferences fall far away from average student characteristics (unrealistic ideals)
- Judgements are narrow

Interactions in the Model

2 per time step – 1 evaluation per interaction

Mandatory e.g. Class environment or group work

Optional e.g. breaks

For each student, a random partner is selected

Positive evaluation → Positive interaction
Negative evaluation→ Negative interaction

Student decides to initiate or not interaction

The random partner decides to accept or refuse the interaction

Refused Interaction

The same interaction can be positive for one student and negative for the other.

We need your feedback

The success of the Bullying model is only possible if we work together!

It is easier to measure exclusion by comparing the positive interactions of each student rather than the negative or refused.

Student Parameters

Student Characteristics:
Extrinsic : directly accessible by all
Intrinsic : learned during interactions
Learning : number of characteristics learned per interaction

Individual Preference:
Informed by Cultural Norms and Individual Judgement

Evaluation: + if a characteristic falls within individual preference
- If it is not within individual preferences

Attraction Link

Attraction Link

Student 1

Student 2

After positive interaction

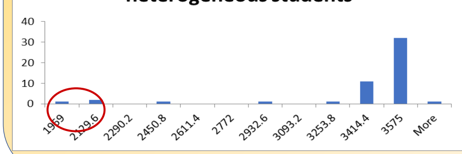
After negative interaction

Attitude: Initial Attraction

Narrow judgement creates the perception of negative interactions and can leave a student feeling marginalised.

It is possible to observe exclusion due to selection of partners

Frequency of positive interactions in heterogeneous students



References

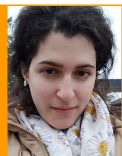
Hofstede, G. H., Hofstede, G. J., & Minkov, M. (2010). Cultures and Organizations : Software for the Mind, Third Edition: Software for the Mind, Third Edition: McGraw-Hill USA.

Sanaria, A. (2004).Conformity and Norms: The individual perspective, Asia Academy of Management (AAOM) Conference,Shanghai,China, December 16-18

Schott, R., & Søndergaard, D. (Eds.). (2014). *School Bullying: New Theories in Context*. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139226707

Thibaut, J. W., & Kelley, H. H. (1959). The social psychology of groups. Oxford, England: John Wiley.

Computer Tools for Modelling Social Conflict Project (CTMSC)



Themis Dimitra Xanthopoulou, PhD
themis.d.xanthopoulou@uia.no
https://www.uia.no/en/kk/profil/themisdx

Generation Rules of ODD2NetLogo

a Concepts from the Folder "4FutureUse"

```
text gen component for concept ForEach {
  (node)->void {
    append indent {ask } ;
    append ${node.collection} { [} \n ;
    with indent {
      append indent {let _} ${node.who.name} { self} \n ;
      append $list{node.activities} ;
    }
    append indent {]} \n ;
  }
}
```

```
text gen component for concept OccupiedSpot {
  (node)->void {
    append {if any? other turtles-here} ;
  }
}
```

b Concepts from the Folder "Actions and Procedures"

```
text gen component for concept Action {
  (node)->void {
    append \n {to} { } ${node.name} { [ _me ] ; } ${node.description}
        \n ;
    append {ask _me [ ; all access is to _me} \n ;
    with indent {
      foreach action in node.activities {
        append ${action} ;
      }
    }
    append {]} \n {end} \n ;
  }
}
```

```
text gen component for concept ActionCall {
  (node)->void {
    boolean wrap = false;
    if (node.parent.isInstanceOf(ProcessOverviewAndScheduling)) {
      wrap = true;
    } else {
      node<Action> myAction = node.ancestor<concept = Action>;
      // wrap = myAction.actor.first:ConcreteEntityReference.generalEntity != node.called.actor.first:ConcreteEntityReference.generalEntity;
    }
    if (wrap) {
      append indent {ask } ;
      if (node.called.actor.isInstanceOf(ConcreteEntityReference)) {
        if (node.called.actor:ConcreteEntityReference.generalEntity.isInstanceOf(EnvironmentEntity)) {
          append {patches} ;
        } else {
          append ${node.called.actor:ConcreteEntityReference.generalEntity.name} {s} ;
        }
      } else {
        append ${node.called.actor} ;
      }
      append { [} \n ;
      increase depth ;
    }

    if (node.called.isInstanceOf(Interaction)) {
      if (wrap) {
        append indent {let _me self} \n ;
      }
      if (node.actuals.size > 1 && !node.actuals[1].isInstanceOf(EntityExpression)) {
        append indent {ask } ${node.actuals[1]} { [} \n ;
        with indent {
          append indent ${node.called.name} { } ${node.actuals[0]} ;
          append { self} \n ;
        }
        append indent {]} \n ;
      } else {
        append indent ${node.called.name} ;
        if (node.actuals.isEmpty) {
          append { _me} ;
        }
        foreach a in node.actuals {
          append { } ${a} ;
```

```
          }
          append \n ;
        }
      } else {
        append indent ${node.called.name} ;
        if (wrap) {
          append { self} ;
        } else {
          append { _me} ;
        }
        foreach a in node.actuals {
          append { } ${a} ;
        }
        append \n ;
      }

      if (wrap) {
        decrease depth ;
        append indent {]} \n ;
      }
    }
}
```

```
text gen component for concept ChangeEnvironment {
  (node)->void {
    append indent {set pcolor } ${node.environment.defaultColour:
        ColourConstant.colour.name} \n ;
  }
}
```

```
text gen component for concept Clone {
  (node)->void {
    append indent {hatch } ${"" + node.amountOfClones} { [} \n ;
    with indent {
      // append indent {rt random-float 360 fd 1} \n ;
      append $list{node.cloneValues} ;
    }
    append indent {]} \n ;
  }
}
```

```
text gen component for concept ConditionalAction {
  (node)->void {
    string ifkind = node.noActivities.isNotEmpty ? "ifelse " : "if ";
    append indent ${ifkind} ${node.condition} { [} \n ;
    with indent {
      foreach action in node.yesActivities {
        append ${action} ;
```

```
      }
    }
    if (node.noActivities.isNotEmpty) {
      append indent {] [ ; else} \n ;
      with indent {
        foreach action in node.noActivities {
          append ${action} ;
        }
      }
    }
    append indent {]} \n ;
  }
}
```

```
text gen component for concept DecrementAttribute {
  (node)->void {
    node<UserDefinedAttribute> uda = ((node<UserDefinedAttribute>)
        node.lhs.attribute);
    if (uda.isInRole(Network : userDefinedAttributes)) {
      if (node.lhs.accessWho.is(me)) {
        append indent {ask out-} ${node.lhs.attribute.parent:Network.
            name} {-to _other [} \n ;
      } else {
        append indent {ask in-} ${node.lhs.attribute.parent:Network.
            name} {-from _other [} \n ;
      }
      increase depth ;
    }
    append indent {set } ${uda.name} {     } ;
    boolean isRange = false;
    string lower = "low";
    string upper = "up";
    if (uda.type.isInstanceOf(RangeType)) {
      node<RangeType> r = uda.type:RangeType;
      isRange = true;
      lower = r.lower;
      upper = r.upper;
    }
    if (isRange) {
      append {max (list } ${lower} { min (list } ${upper} { (} ;
    }
    append ${node.lhs.attribute.name} { - } ${node.rhs} ;
    if (isRange) {
      append {)))} ;
    }
    append \n ;
    if (uda.isInRole(Network : userDefinedAttributes)) {
```

```
          decrease depth ;
          append indent {]} \n ;
        }
        // thisshouldbeamethodforallthesetters
        if (uda.isEnvironmentAttribute()) {
          foreach envEntity in node.containingRoot.descendants<concept = EnvironmentEntity> {
            ifInstanceOf (envEntity.defaultColour is ScaledColour scaled) {
              if (scaled.scalingValue.attribute == uda) {
                append indent {set pcolor } ${scaled} \n ;
              }
            }
          }
        }
    }
}
```

```
text gen component for concept Function {
  (node)->void {
    append \n {to-report } ${node.name} { [ } ;
    foreach param in node.parameters {
      if (param.type.isInstanceOf(ConcreteEntityReference)) {
        append {_} ;
      }
      append ${param.name} { } ;
    }
    append {]} { ; } ${node.description} \n ;
    with indent {
      foreach let in node.locals {
        append ${let} ;
      }
      append indent {report } ${node.result} \n ;
    }
    append {end} \n ;
  }
}
```

```
text gen component for concept IncrementAttribute {
  (node)->void {
    node<UserDefinedAttribute> uda = (node<UserDefinedAttribute>) (node.lhs.attribute);
    if (uda.isInRole(Network : userDefinedAttributes)) {
      if (node.lhs.accessWho.is(me)) {
        append indent {ask out-} ${node.lhs.attribute.parent:Network.name} {-to _other [} \n ;
      } else {
```

```
        append indent {ask in—} ${node.lhs.attribute.parent:Network.
          name} {—from _other [} \n ;
      }
      increase depth ;
    }
    append indent {set } ${uda.name} {     } ;
    boolean isRange = false;
    string lower = "low";
    string upper = "up";
    if (uda.type.isInstanceOf(RangeType)) {
      node<RangeType> r = uda.type:RangeType;
      isRange = true;
      lower = r.lower;
      upper = r.upper;
    }
    if (isRange) {
      append {max (list } ${lower} { min (list } ${upper} { (} ;
    }
    if (uda.type.isInstanceOf(Set)) {
      append {remove—duplicates (sentence } ${node.lhs.attribute.name
        } { } ${node.rhs} {)} ;
    } else {
      append ${node.lhs.attribute.name} { + } ${node.rhs} ;
    }
    if (isRange) {
      append {)))} ;
    }
    append \n ;
    if (uda.isInRole(Network : userDefinedAttributes)) {
      decrease depth ;
      append indent {]} \n ;
    }
    // thisshouldbeamethodforallthesetters
    if (uda.isEnvironmentAttribute()) {
      foreach envEntity in node.containingRoot.descendants<concept =
        EnvironmentEntity> {
        ifInstanceOf (envEntity.defaultColour is ScaledColour scaled)
          {
          if (scaled.scalingValue.attribute == uda) {
            append indent {set pcolor } ${scaled} \n ;
          }
        }
      }
    }
  }
}
```

```
text gen component for concept Interaction {
  (node)->void {
    boolean partner2isEnv = false;
    string partner2ColourCheck = "dummy";
    ifInstanceOf (node.partner2 is ConcreteEntityReference ref) {
      if (ref.generalEntity.isInstanceOf(EnvironmentEntity)) {
        partner2isEnv = true;
        ifInstanceOf (ref.generalEntity:EnvironmentEntity.
            defaultColour is ColourConstant constant) {
          partner2ColourCheck = "pcolor = " + constant.colour.name;
        }
        ifInstanceOf (ref.generalEntity:EnvironmentEntity.
            defaultColour is ScaledColour constant) {
          partner2ColourCheck = "shade-of? pcolor " + constant.colour
              .name;
        }
      }
    }
    ifInstanceOf (node.partner2 is EnvironmentEntityReference ref) {
      partner2isEnv = true;
      ifInstanceOf (ref.envEntity.defaultColour is ColourConstant
          constant) {
        partner2ColourCheck = "pcolor = " + constant.colour.name;
      }
      ifInstanceOf (ref.envEntity.defaultColour is ScaledColour
          constant) {
        partner2ColourCheck = "shade-of? pcolor " + constant.colour.
            name;
      }
    }
    if (partner2isEnv) {
      append \n {to } ${node.name} { [ _me ] ; } ${node.description}
          \n ;
      append {ask _me [ ; all access is to _me} \n ;
      with indent {
        append indent {if } ${partner2ColourCheck} { [} \n ;
        with indent {
          append $list{node.activities} ;
        }
        append indent {]} \n ;
      }
    } else {
      string otherString = node.partner2.isInstanceOf(AnyEnvironment)
          ? "" : "_other ";
      append \n {to } ${node.name} { [ _me } ${otherString} {] ; } ${
          node.description} \n ;
      append {ask _me [ ; all access is to _me} \n ;
```

```
      with indent {
        append $list{node.activities} ;
      }
    }
    append {]} \n {end} \n ;
  }
}
```

```
text gen component for concept KillEntity {
  (node)->void {
    if (node.killWho.is(me)) {
      append indent {die} \n ;
    } else {
      append indent {ask _other} { [ die ]} \n ;
    }
  }
}
```

```
text gen component for concept LetActivity {
  (node)->void {
    append indent {let } ${node.name} { } ${node.value} \n ;
  }
}
```

```
text gen component for concept Move {
  (node)->void {
    enummember<Direction> hDir = node.dir;
    if (node.dir.getOrdinal() > enum/Direction/.random.getOrdinal())
      {
      hDir = enum/Direction/.members.get(hDir.getOrdinal() - enum/
         Direction/.aheadRandom.getOrdinal());
    }
    enum switch (node.dir) {
      ahead, aheadRandom, backwards, backwardsRandom -> {
        // nothingneededhere
      }
      right, rightRandom -> {
        append indent {right 90} \n ;
      }
      left, leftRandom -> {
        append indent {left 90} \n ;
      }
      back, backRandom -> {
        append indent {right 180} \n ;
      }
      north, northRandom -> {
        append indent {set heading 0} \n ;
      }
```

```
      south, southRandom -> {
        append indent {set heading 180} \n ;
      }
      east, eastRandom -> {
        append indent {set heading 90} \n ;
      }
      west, westRandom -> {
        append indent {set heading 270} \n ;
      }
      random -> {
        append indent {right random 360} \n ;
      }
    };
    if (node.dir.getOrdinal() > enum/Direction/.random.getOrdinal())
       {
      // wigglesomewhataroundthechosendirection
      append indent {right random 90} \n ;
      append indent {left random 90} \n ;
    }
    if (node.dir.is(backwards) || node.dir.is(backwardsRandom)) {
      append indent {back } ${node.distance} \n ;
    } else {
      append indent {forward } ${node.distance} \n ;
    }
    append indent {setxy pxcor pycor ; align to center of patch} \n ;
  }
}
```

```
text gen component for concept SetAttribute {
  (node)->void {
    node<UserDefinedAttribute> uda = (node<UserDefinedAttribute>) (
       node.lhs.attribute);
    if (uda.isInRole(Network : userDefinedAttributes)) {
      if (node.lhs.accessWho.is(me)) {
        append indent {ask out-} ${node.lhs.attribute.parent:Network.
           name} {-to _other [} \n ;
      } else {
        append indent {ask in-} ${node.lhs.attribute.parent:Network.
           name} {-from _other [} \n ;
      }
      increase depth ;
    }
    append indent {set } ${uda.name} {     } ;
    boolean isRange = false;
    string lower = "low";
    string upper = "up";
    if (uda.type.isInstanceOf(RangeType)) {
```

```
        node<RangeType> r = uda.type:RangeType;
        isRange = true;
        lower = r.lower;
        upper = r.upper;
      }
      if (isRange) {
        append {max (list } ${lower} { min (list } ${upper} { (} ;
      }
      append ${node.rhs} ;
      if (isRange) {
        append {))))} ;
      }
      append \n ;
      if (uda.isInRole(Network : userDefinedAttributes)) {
        decrease depth ;
        append indent {]} \n ;
      }
      // thisshouldbeamethodforallthesetters
      if (uda.isEnvironmentAttribute()) {
        foreach envEntity in node.containingRoot.descendants<concept =
            EnvironmentEntity> {
          ifInstanceOf (envEntity.defaultColour is ScaledColour scaled)
              {
            if (scaled.scalingValue.attribute == uda) {
              append indent {set pcolor } ${scaled} \n ;
            }
          }
        }
      }
    }
  }
```

c Concepts from the Folder “Appearence”

```
text gen component for concept ColourConstant {
  (node)->void {
    append ${"" + node.colour} ;
  }
}
```

```
text gen component for concept ScaledColour {
  (node)->void {
    // somebadtrick-needtobemademoregeneral
    append {scale-color } ${node.colour.name} { (10 - } ${node.
        scalingValue.attribute.name} {) -10 20} ;
  }
```

```
}
```

d Concepts from the Folder "EntitiesAndAttributes"

```
text gen component for concept AnyEntity {
  (node)->void {
    append {turtles} ;
  }
}
```

```
text gen component for concept AnyEnvironment {
  (node)->void {
    append {patches} ;
  }
}
```

```
text gen component for concept ConcreteEntityReference {
  (node)->void {
    append ${node.generalEntity.name} {s} ;
  }
}
```

```
text gen component for concept Entity {
  (node)->void {
    append {breed[ } ${node.name} { a-} ${node.name} { ]} \n ;
    foreach e in node.userDefinedAttributes {
      append ${node.name} {-own} {[ } ${e.name} { ]} \n ;
    }
  }
}
```

e Concepts from the Folder "Experiments"

```
text gen component for concept AttributeSampling {
  (node)->void {
    concept switch (node.method.concept) {
      exactly FixedValue :
        append indent {<enumeratedValueSet variable="} ${node.
            attribute.name} {">} \n ;
        with indent {
          append ${node.method} ;
        }
        append indent {</enumeratedValueSet>} \n ;
      exactly RangeSampling :
```

```
        append indent {<steppedValueSet variable="} ${node.attribute.name} {" } ${node.method} {/>} \n ;
      default :
        append {sampling method not implemented yet} ;
    }
  }
}
```

```
text gen component for concept EntitySampling {
  (node)->void {
    concept switch (node.method.concept) {
      exactly FixedValue :
        append indent {<enumeratedValueSet variable="num-} ${node.entity.name} {s">} \n ;
        with indent {
          append ${node.method} ;
        }
        append indent {</enumeratedValueSet>} \n ;
      exactly RangeSampling :
        append indent {<steppedValueSet variable="num-} ${node.entity.name} {s" } ${node.method} {/>} \n ;
      default :
        append {sampling method not implemented yet} ;
    }
  }
}
```

```
text gen component for concept ExperimentDefinition {
  (node)->void {
    append indent {<experiment name="} ${node.name} {" repetitions="} ${"" + node.repetitions} {" runMetricsEveryStep="false">} \n ;
    with indent {
      append indent {<setup>setup</setup>} \n ;
      append indent {<go>go</go>} \n ;
      nlist<TimedCondition> ends = node.descendants<concept = TimedCondition>;
      if (ends.isNotEmpty) {
        append indent {<timeLimit steps="} ${ends.first.value} {"/>} \n ;
      }
      append csvgencalls node.dataCollection ;
      foreach c in node.dataCollection.collectItem {
        append indent {<metric>} ${c.method.name} { } ${c.TODO_value} {</metric>} \n ;
      }
      append $list{node.experimentValues} ;
    }
```

```
    append indent {</experiment>} \n ;
  }
}
```

```
text gen component for concept FixedValue {
  (node)->void {
    append indent {<value value="} ${node.value} {"/>} \n ;
  }
}
```

```
text gen component for concept GraphicsWindow {
  (node)->void {
    <no statements>
  }
}
```

```
text gen component for concept RangeSampling {
  (node)->void {
    append {first="} ${node.lowerBound} {" step="} ${node.increment} {" last="} ${node.upperBound} {"} ;
  }
}
```

```
text gen component for concept TODO_DataCollection {
  (node)->void {
    sequence<node<Entity>> entities = node.containingRoot:ODD.entitiesAndVariables.entities;
    string generalEntityName = entities.size == 1 ? entities.first.name : "turtle";

    sequence<node<AttributeAccess>> networkArrayAttributes = node.REMOVE_collect.where({~it => it.attribute.isNetworkAttribute() && ((node<UserDefinedAttribute>) (it.attribute)).type.isInstanceOf(Array); });
    if (networkArrayAttributes.size != 0) {
      append {; TO BE DONE creating network array attributes} \n ;
      foreach c in networkArrayAttributes {
        append {; making new collection for array attribute in network: } ${c.attribute.name} \n ;
      }
    }

    sequence<node<AttributeAccess>> networkOtherAttributes = node.REMOVE_collect.where({~it => it.attribute.isNetworkAttribute() && !((node<UserDefinedAttribute>) (it.attribute)).type.isInstanceOf(Array); });
    if (networkOtherAttributes.size != 0) {
```

```
      append csvgen true , networkOtherAttributes . first . attribute .
         parent : Network . name , networkOtherAttributes , new arraylist <
         node<TODO_LetCollect>> ;
    }

    sequence<node<AttributeAccess>> agentArrayAttributes = node .
       REMOVE_collect . where({~ it => ! it . attribute . isNetworkAttribute
       () && ((node<UserDefinedAttribute>) ( it . attribute ) ) . type .
       isInstanceOf ( Array ) ; }) ;
    if ( agentArrayAttributes . size != 0) {
      foreach c in agentArrayAttributes {
        string attributeName = c . attribute . isInRole ( Entity :
           userDefinedAttributes ) ? c . attribute . parent : Entity . name :
           generalEntityName ;
        append csvgen false , attributeName , new arraylist <node<
           AttributeAccess>>{c } , new arraylist <node<TODO_LetCollect>>
            ;
      }
    }

    sequence<node<AttributeAccess>> agentOtherAttributes = node .
       REMOVE_collect . where({~ it => ! it . attribute . isNetworkAttribute
       () && !((node<UserDefinedAttribute>) ( it . attribute ) ) . type .
       isInstanceOf ( Array ) ; }) ;
    if ( agentOtherAttributes . size != 0) {
      foreach entity in entities {
        sequence<node<AttributeAccess>> myAttributes =
           agentOtherAttributes . where({~ it => it . attribute . parent ==
           entity || it . attribute . parent . isInstanceOf (
           EntitiesStateVariablesAndScales ) ; }) ;
        append csvgen false , entity . name , myAttributes , node .
           REMOVE_collectsyn ;
      }
    }
  }
}
```

f Concepts from the Folder “Experiments”

f.1 Concepts from the Subfolder “Condition”

```
text gen component for concept BooleanConstant {
  (node)->void {
    if (node . value ) {
      append ${" True"} ;
    } else {
```

```
      append ${"False"} ;
    }
  }
}
```

```
text gen component for concept Comparison {
  (node)->void {
    append ${node.left} ;
    append { } ${node.operator.presentation} { } ;
    append ${node.right} ;
  }
}
```

```
text gen component for concept QuantifiedCondition {
  (node)->void {
    if (node.entity.isInstanceOf(Entity)) {
      if (node.quantifier.is(NONE)) {
        append indent {if not any? } ${node.entity.name} {s [ stop ]}
            \n ;
      }
      if (node.quantifier.is(ALL)) {
        append indent {if count turtles = count} ${node.entity.name}
            {s [ stop ]} \n ;
      }
    } else {
      string comparison = " ??? ";
      if (node.quantifier.is(ALL)) {
        comparison = " = ";
      }
      if (node.quantifier.is(NONE)) {
        comparison = " != ";
      }
      append indent {if all? patches [pcolor} ${comparison} ${node.
          entity:EnvironmentEntity.defaultColour:ColourConstant.colour
          .name} {] [ calculate stop ]; conditioned end of simulation}
          \n ;
    }
  }
}
```

```
text gen component for concept QuantifiedConditionWithComparison {
  (node)->void {
    if (node.entity.isNull) {
      if (node.quantifier.is(NONE)) {
        append indent {if not any? turtle [ } ${node.comp} { ] [ stop
            ]} \n ;
      }
      if (node.quantifier.is(ALL)) {
```

```
        append indent {if all? turtles [ } ${node.comp} { ] [ stop ]}
            \n ;
      }

    } else {
      append {???? not implemented yet: quantified condition ????} ;
    }
  }
}
```

```
text gen component for concept TimedCondition {
  (node)->void {
    append indent {if ticks } ${node.operator.presentation} { } ${
        node.value} { [ calculate stop ] ; timed end of simulation} \n
        ;
  }
}
```

```
text gen component for concept TODO_ConditionWithSame {
  (node)->void {
    append {with [ } ${node.baseEntityAttributes.name} { } ${node.
        comparison.presentation} { } ;

    append {[ } ${node.baseEntityAttributes.name} { ] } {of myself ]}
        ;
  }
}
```

f.2 Concepts from the pool of the folder “Experiments”

```
text gen component for concept AttributeAccess {
  (node)->void {
    if (node.attribute.isNetworkAttribute()) {
      if (node.accessWho.is(me)) {
        append {[} ${node.attribute.name} {] of out-} ${node.
            attribute.parent:Network.name} {-to _other} ;
      } else {
        append {[} ${node.attribute.name} {] of in-} ${node.attribute
            .parent:Network.name} {-from _other} ;
      }
    } else if ((node.attribute.isEntityAttribute()) && node.accessWho
        .is(other)) {
      append {[} ${node.attribute.name} {] of _other} ;
    } else {
      append ${node.attribute.name} ;
    }
  }
```

```
}
```

```
text gen component for concept Binary {
  (node)->void {
    if (node.operator.is(distance)) {
      append {abs(} ${node.left} { - } ${node.right} {)} ;
    } else {
      append ${node.left} { } ${node.operator.presentation} { } ${
          node.right} ;
    }
  }
}
```

```
text gen component for concept Collect {
  (node)->void {
    ifInstanceOf (node.inner is AttributeAccess attrAcc) {
      if (attrAcc.attribute.isInRole(Network : userDefinedAttributes)
          ) {
        string networkName = attrAcc.attribute.parent:Network.name;
        string direction = attrAcc.accessWho.is(me) ? "out-" : "in-";
        string endDirect = attrAcc.accessWho.is(me) ? "-to" : "-from
            ";
        string calculation = node.kind.is(StdDev) ? "Standard-
            deviation" : node.kind.name;
        append {precision (} ${calculation} { (map } ;
        append {[ _o -> [ } ${attrAcc.attribute.name} { ] of _o ]} ;
        append { [ self ] of my-} ${direction} ${networkName} {s)) 2}
            ;
        return;
      }
    }
    ifInstanceOf (node.inner is LetAccess letAcc) {
      if (letAcc.let.isInRole(EntitiesStateVariablesAndScales :
          syntheticAttributes)) {
        append ${node.kind.name} { [} ${node.inner} {] of turtles} ;
        return;
      }
    }
    append ${node.kind.name} { } ${node.inner} ;
  }
}
```

```
text gen component for concept ElementWise {
  (node)->void {
    append {(map} \n ;
    with indent {
      with indent {
        concept switch (node.inner.concept) {
```

```
            exactly Binary :
              enummember<BinaryOperator> op = node.inner:Binary.
                  operator;
              string operation = op.is(distance) ? "abs (v1 - v2)" : "
                  v1 " + op.presentation + " v2";
              append indent {[ [ v1 v2 ] -> } ${operation} { ]} \n ;
              append indent ${node.inner:Binary.left} \n ;
              append indent ${node.inner:Binary.right} \n ;
            exactly IfThenElse :
              // thefollowingselectionshouldberelatedtotyping(arraytype
                  )
              node<LetAccess> iterationVariable = node.inner.
                  descendants<concept = LetAccess>.first;
              append indent {[ } ${iterationVariable} { -> } ${node.
                  inner} { ]} \n ;
              append indent ${iterationVariable.let.name} \n ;
            default :
              error "unknown expression kind in ElementWise: " + node.
                  inner.concept ;
          }
        }
        append indent {)} ;
      }
    }
}
```

```
text gen component for concept Empty {
  (node)->void {
    append {[]} ;
  }
}
```

```
text gen component for concept EntityAccess {
  (node)->void {
    ifInstanceOf (node.parent is ForEach fe) {
      if (fe.who.is(other)) {
        append {other } ;
      }
    }
    append ${node.entity} ;
  }
}
```

```
text gen component for concept EntityExpression {
  (node)->void {
    if (node.who.is(me)) {
      ifInstanceOf (node.parent is ActionCall call) {
        if (!call.called.isInstanceOf(Interaction)) {
```

```
          append {self} ;
          return;
        }
      }
    }
    append {_} ${node.who.name} ;
  }
}
```

```
text gen component for concept False {
  (node)->void {
    append {false} ;
  }
}
```

```
text gen component for concept FunctionCall {
  (node)->void {
    append ${node.function.name} { } ;
    foreach actual in node.actuals {
      append ${actual} { } ;
    }
  }
}
```

```
text gen component for concept IfThenElse {
  (node)->void {
    append {ifelse-value } ${node.condition} { [ } ${node.positive} {
        ] [ } ${node.negative} { ]} ;
  }
}
```

```
text gen component for concept Indices {
  (node)->void {
    append {range length (} ${node.inner} {)} ;
  }
}
```

```
text gen component for concept LetAccess {
  (node)->void {
    node<ElementWise> ew = node.ancestor<concept = ElementWise>;
    if (ew != null && ew.inner.isInstanceOf(IfThenElse)) {
      append {_} ;
    }
    append ${node.let.name} ;
  }
}
```

```
text gen component for concept NumberConstant {
```

```
  (node)->void {
    append ${"" + node.value} ;
  }
}
```

```
text gen component for concept Parenthesis {
  (node)->void {
    append {(} ${node.inner} {)} ;
  }
}
```

```
text gen component for concept Percentage {
  (node)->void {
    append ${node.value} ;
  }
}
```

```
text gen component for concept SelectN {
  (node)->void {
    string other = "";
    if (node.parent.isInstanceOf(ForEach) && node.parent:ForEach.who.
       is(other)) {
      other = "other ";
    }
    if (node.parent.isInstanceOf(ActionCall)) {
      other = "other ";
    }
    if (node.count.isInstanceOf(Percentage)) {
      append {up-to-n-of ((num-} ${node.argument} { * } ${node.count}
          {) / 100) } ${other} ${node.argument} ;
    } else {
      append {up-to-n-of } ${node.count} { } ${other} ${node.argument
         } ;
    }
    if (node.where.is(sameSpot)) {
      append {-here} ;
    }
  }
}
```

```
text gen component for concept TODO_CountAllEntities {
  (node)->void {
    append {count } ${node.what} { with [} ${node.criteria} {]} ;
  }
}
```

```
text gen component for concept TODO_CountEntities {
  (node)->void {
```

```
      append {count } ;
      if (node.where.is(all)) {
        append {turtles} ;
      }
      if (node.where.is(neighbour)) {
        append {(turtles-on neighbors)} ;
      }
      if (node.where.is(same)) {
        append {turtles-here} ;
      }
      if (node.countCondition != null) {
        append { } ${node.countCondition} ;
      }
    }
  }
```

```
text gen component for concept TODO_FilterIndex {
  (node)->void {
    append {(map} \n ;
    with indent {
      with indent {
        append indent {[ v -> item v } ${node.argument} { ]} \n ;
        append indent ${node.filterCondition} \n ;
      }
      append indent {)} ;
    }
  }
}
```

```
text gen component for concept True {
  (node)->void {
    append {true} ;
  }
}
```

g Concepts from the Folder “Initialization”

```
text gen component for concept Area {
  (node)->void {
    ifInstanceOf (node.ancestor<concept = EnvironmentEntity> is
        EnvironmentEntity environmentEntity) {
      if (node.area.is(left)) {
        append indent {if pxcor = } {min-pxcor[} \n ;
      }
      if (node.area.is(right)) {
        append indent {if pxcor = } {max-pxcor[} \n ;
```

```
      }
      if (node.area.is(top)) {
        append indent {if pycor = } {max-pycor[} \n ;
      }
      if (node.area.is(bottom)) {
        append indent {if pycor = } {min-pycor[} \n ;
      }
      with indent {
        append indent {set pcolor } ${environmentEntity.defaultColour
            :ColourConstant.colour.name} \n ;
      }
      append indent {]} \n ;
    }
  }
}
```

```
text gen component for concept AttributeInit {
  (node)->void {
    if (node.initialisation == null) {
      append indent {set } ${node.attribute.name} { 0 ; no
          initialization defined} \n ;
      return;
    }
    ifInstanceOf (node.initialisation is Slider slider) {
      return;
    }
    append indent {set } ${node.attribute.name} { } ;
    boolean isArray = node.attribute.type.isInstanceOf(Array) && !
        node.initialisation.isInstanceOf(Empty);
    if (isArray) {
      append {n-values } ${node.attribute.type:Array.size} { [ } ;
    }
    boolean isRange = false;
    string lower = "low";
    string upper = "up";
    if (node.attribute.type.isInstanceOf(RangeType)) {
      node<RangeType> r = node.attribute.type:RangeType;
      isRange = true;
      lower = r.lower;
      upper = r.upper;
    } else if (node.attribute.type.isInstanceOf(Array) && node.
        attribute.type:Array.inner.isInstanceOf(RangeType)) {
      node<RangeType> r = node.attribute.type:Array.inner:RangeType;
      isRange = true;
      lower = r.lower;
      upper = r.upper;
    }
```

```
      if (isRange) {
        append {max (list } ${lower} { min (list } ${upper} { (} ;
      }
      append ${node.initialisation} ;
      if (isRange) {
        append {))))} ;
      }
      if (isArray) {
        append { ]} ;
      }
      append \n ;
    }
  }
```

```
text gen component for concept Density {
  (node)->void {
    ifInstanceOf (node.ancestor<concept = EnvironmentEntity> is
        EnvironmentEntity envEntity) {
      ifInstanceOf (node.percentage is Slider slider) {
        append indent {if (random 100) < } ${slider.generatedName}
            {[} \n ;
      }
      ifInstanceOf (node.percentage is NumberConstant numConstant) {
        append indent {if (random 100) < } ${"" + numConstant.value}
            {[} \n ;
      }
      with indent {
        append indent {set pcolor } ${envEntity.defaultColour:
            ColourConstant.colour.name} \n ;
      }
      append indent {]} \n ;
    }
  }
}
```

```
text gen component for concept Everywhere {
  (node)->void {
    ifInstanceOf (node.ancestor<concept = EnvironmentEntity> is
        EnvironmentEntity envEntity) {
      append indent {set pcolor } ${envEntity.defaultColour} \n ;
    }
  }
}
```

```
text gen component for concept Random {
  (node)->void {
    append {setxy random-xcor random-ycor ; spread } ${node.ancestor<
        concept = INamedConceptODD>.name} {s throughout the
```

```
        environment} \n ;
    }
}
```

```
text gen component for concept RandomNormal {
  (node)->void {
    append {precision (} ;
    append {random-normal } ${node.mean} { } ${node.stdev} ;
    append {) 2} ;
    // fixedprecision2forthemoment
  }
}
```

```
text gen component for concept RandomUniform {
  (node)->void {
    append {precision (} ;
    if (node.from.isInstanceOf(NumberConstant) && node.from:
        NumberConstant.value.equals("0")) {
      append {random-float (} ${node.to} {)} ;
    } else {
      append ${node.from} { + random-float ((} ${node.to} {) - (} ${
          node.from} {))} ;
    }
    append {) 2} ;
    // fixedprecision2forthemoment
  }
}
```

```
text gen component for concept Slider {
  (node)->void {
    // namecouldbegeneratedheredirectlyandnotinconstraints
    append ${node.generatedName} \n ;
    append ${node.generatedName} \n ;
    append ${"" + node.minAmount} \n ;
    append ${"" + node.maxAmount} \n ;
    append ${"" + node.startValue} \n ;
    append ${"" + node.slideAmount} \n ;
  }
}
```

h Concepts from the pool of concepts

```
text gen component for concept EntitiesStateVariablesAndScales {
  (node)->void {
    string defaultColor = "red";
    string distributionDummy = "dummy";
```

```
    foreach p in node.environmentEntities {
      append {###### This is still to be done!} ;
      append {if pcolor = } ${p.defaultColour:ColourConstant.colour.name} {[} \n ;
      append {]} \n ;
    }

    foreach p in node.environmentEntities {
      if (distributionDummy :eq: "random") {
        append {if (random 100) < } ${p.name} {slider[} \n ;
        append {set pcolor } ${p.defaultColour:ColourConstant.colour.name} \n ;
        append {]} \n ;
      }
      if (distributionDummy :eq: "left") {
        append {if pxcor = } {min-pxcor[} \n ;
        append {set pcolor } ${p.defaultColour:ColourConstant.colour.name} \n ;
        append {]} \n ;
      }
      if (distributionDummy :eq: "right") {
        append {if pxcor = } {max-pxcor[} \n ;
        append {set pcolor } ${p.defaultColour:ColourConstant.colour.name} \n ;
        append {]} \n ;
      }
      if (distributionDummy :eq: "bottom") {
        append {if pxcor = } {min-pycor[} \n ;
        append {set pcolor } ${p.defaultColour:ColourConstant.colour.name} \n ;
        append {]} \n ;
      }
      if (distributionDummy == "top") {
        append {if pxcor = } {max-pycor[} \n ;
        append {set pcolor } ${p.defaultColour:ColourConstant.colour.name} \n ;
        append {]} \n ;
      }
    }

    foreach e in node.environmentEntities {
      if (e.name == "BackGround") {
        append {ask patches[ set pcolor } ${defaultColor} { ]} ;
      }
    }
    append {ask patches [} \n ;
```

```
      append {]} \n ;
    }
  }
```

```
text gen component for concept Experiments {
  (node)->void {
    if (node.experiments.isEmpty) { return; }
    append {<experiments>} \n ;
    with indent {
      append $list{node.experiments} ;
    }
    append {</experiments>} \n ;
  }
}
```

```
text gen component for concept ODD {
file name : <Node.name>
  (node)->void {
    foreach r in node.entitiesAndVariables.networks {
      if (!r.directed) {
        append {un} ;
      }
      append {directed-link-breed [} ${r.name} {s } ${r.name} {] ; }
          ${r.description} \n ;
      append ${r.name} {s-own [} \n ;
      with indent {
        foreach uda in r.userDefinedAttributes {
          append indent ${uda.name} { ; } ${uda.description} \n ;
        }
      }
      append {]} \n \n ;
    }

    append {globals [\n} ;
    with indent {
      append indent {__INTERNAL__stop ; variable to indicate that
          there is a stop\n} ;
      foreach g in node.entitiesAndVariables.attributesInit {
        if (!g.initialisation.isInstanceOf(Slider) && g.attribute.
            isInRole(EntitiesStateVariablesAndScales : modelParameters
            )) {
          append indent ${g.attribute.name} { ; } ${g.attribute.
              description} \n ;
        }
      }
      foreach s in node.entitiesAndVariables.syntheticModelAttributes
          {
```

```
        append indent ${s.name} { ; } ${s.description} \n ;
      }
    }
    append {]} \n \n ;

    foreach e in node.entitiesAndVariables.entities {
      append {breed [ } ${e.name} {s } ${e.name} { ]} { ; } ${e.description} \n ;
      if (e.userDefinedAttributes.isNotEmpty) {
        append ${e.name} {-own[} \n ;
        foreach uda in e.userDefinedAttributes {
          append indent ${uda.name} { ; } ${uda.description} \n ;
        }
        append {]} \n ;
      }
      append \n ;
    }

    if (node.entitiesAndVariables.entityAttributes.isNotEmpty || node.entitiesAndVariables.syntheticAttributes.isNotEmpty) {
      append {turtles-own [} \n ;
      with indent {
        foreach a in node.entitiesAndVariables.entityAttributes {
          append indent ${a.name} { ; } ${a.description} \n ;
        }
        foreach s in node.entitiesAndVariables.syntheticAttributes {
          append indent ${s.name} { ; } ${s.description} \n ;
        }
      }
      append {]} \n \n ;
    }

    if (node.entitiesAndVariables.environmentAttributes.isNotEmpty) {
      append {patches-own [} \n ;
      with indent {
        foreach uda in node.entitiesAndVariables.environmentAttributes {
          append indent ${uda.name} { ; } ${uda.description} \n ;
        }
      }
      append {]} \n \n ;
    }

    append {to setup} \n ;
    increase depth ;
    append indent {clear-all} \n ;
    append indent {set __INTERNAL__stop false\n} ;
```

```
    if (node.entitiesAndVariables.environmentEntities.isNotEmpty || node.entitiesAndVariables.environmentAttributes.isNotEmpty) {
      append indent {ask patches [} \n ;
      with indent {
        foreach udainit in node.entitiesAndVariables.attributesInit {
          if (udainit.attribute.isInRole(EntitiesStateVariablesAndScales : environmentAttributes)) {
            append ${udainit} ;
          }
        }
        foreach envEntity in node.entitiesAndVariables.environmentEntities {
          if (envEntity.initialisationLocation != null) {
            append ${envEntity.initialisationLocation} ;
          }
        }
      }
      append indent {]} \n ;
    }

    foreach udainit in node.entitiesAndVariables.attributesInit {
      if (udainit.attribute.isInRole(EntitiesStateVariablesAndScales : modelParameters)) {
        append ${udainit} ;
      }
    }

    foreach entity in node.entitiesAndVariables.entities {
      if (entity.initialisationNumberOfEntities.isNull) { continue; }
      string defaultShape = "turtle";
      string defaultColor = "grey";
      string defaultSize = "1";
      // whyinitwhenweoverwriteimmediately?
      defaultShape = entity.shape.shape.presentation;
      defaultColor = entity.colour:ColourConstant.colour.name;
      defaultSize = entity.shape.size;

      if (!entity.initialisationNumberOfEntities.isInstanceOf(Slider)) {
        append indent {let num-} ${entity.name} {s } ${entity.initialisationNumberOfEntities} \n ;
      }
      append indent {create-} ${entity.name} {s num-} ${entity.name} {s [ ; create the } ${entity.name} {s} \n ;
      with indent {
        foreach udainit in entity.initAttributes {
```

```
        append ${udainit} ;
      }
      foreach udainit in node.entitiesAndVariables.attributesInit {
        if (udainit.attribute.isInRole(EntitiesStateVariablesAndScales : entityAttributes)) {
          append ${udainit} ;
        }
      }
      foreach link in node.entitiesAndVariables.networks.where({~it => it.from == entity; }) {
        append indent {create-} ${link.name} {s-to } ${link.to == link.from ? "other " : ""} ${link.to.name} {s [ ; create relationships} \n ;
        foreach linkUDAinit in node.entitiesAndVariables.attributesInit.where({~it => it.attribute.isInRole(Network : userDefinedAttributes); }) {
          with indent {
            append ${linkUDAinit} ;
          }
        }
        append indent {]} \n ;
      }
    }

    foreach entityUdaInit in entity.initAttributes {
      if (entityUdaInit.initialisation != null) {
        with indent {
          append indent {set } ${entityUdaInit.attribute.name} { } ;
          if (entityUdaInit.initialisation.isInstanceOf(Slider)) {
            append ${entityUdaInit.initialisation:Slider.generatedName} ;
          } else {
            append ${entityUdaInit.initialisation} ;
          }
          append \n ;
        }
      }
    }


    // Color
    with indent {
      append indent {set color } ${defaultColor} \n ;
      append indent {set shape "} ${defaultShape} ${"\""} \n ;
      append indent {set size } ${"" + defaultSize} \n ;
      if (entity.initialisationDistribution != null) {
```

```
          append indent ${entity.initialisationDistribution} ;
        }
      }
      append indent {]} \n ;
    }
    if (node.entitiesAndVariables.syntheticAttributes.isNotEmpty) {
      append indent {calculate ; init synthetic attributes} \n ;
    }
    if (node.entitiesAndVariables.syntheticModelAttributes.isNotEmpty) {
      append indent {calculate-globals ; init synthetic model attributes} \n ;
    }
    append indent {reset-ticks} \n ;
    decrease depth ;
    append {end} \n ;

    if (node.processAndScheduling != null) {
      append ${node.processAndScheduling} ;
    }


    append {to stopping\n} ;
    with indent {
      append indent {set __INTERNAL__stop true\n} ;
    }
    append {end} \n ;

    if (node.entitiesAndVariables.syntheticAttributes.isNotEmpty) {
      append \n {to calculate; compute synthetic attributes} \n ;
      with indent {
        append indent {ask turtles [} \n ;
        with indent {
          foreach syn in node.entitiesAndVariables.syntheticAttributes {
            append indent {set } ${syn.name} { } ${syn.value} \n ;
          }
        }
        append indent {]} \n ;
      }
      append {end} \n ;
    }

    if (node.entitiesAndVariables.syntheticModelAttributes.isNotEmpty) {
      append \n {to calculate-globals; compute synthetic model attributes} \n ;
```

```
      with indent {
        foreach syn in node.entitiesAndVariables.syntheticModelAttributes {
          append indent {set } ${syn.name} { } ${syn.value} \n ;
        }
      }
      append {end} \n ;
    }


    append {\n; Automatically generated by ODD2NetLogo.\n; See Info tab for more information.\n} ;
    append {@#$#@#$#@} \n ;
    append {GRAPHICS-WINDOW} \n ;
    append {250} \n {10} \n {100} \n {100} \n {-1} \n {-1} \n ;
    if (node.experiments.defaultWorld.pixelSize :ne: 0) {
      append ${"" + node.experiments.defaultWorld.pixelSize} \n ;
    } else {
      append {2} \n ;
    }
    append {1} \n {10} \n {1} \n {1} \n {1} \n {0} \n ;
    if (node.experiments.defaultWorld.wrapHorizontal) {
      append {1} \n ;
    } else {
      append {0} \n ;
    }
    if (node.experiments.defaultWorld.wrapVertical) {
      append {1} \n ;
    } else {
      append {0} \n ;
    }
    append {1} \n ;
    if (node.experiments.defaultWorld.worldSize :ne: 0) {
      append {-} ${"" + node.experiments.defaultWorld.worldSize} \n ;
      append ${"" + node.experiments.defaultWorld.worldSize} \n ;
      append {-} ${"" + node.experiments.defaultWorld.worldSize} \n ;
      append ${"" + node.experiments.defaultWorld.worldSize} \n ;
    } else {
      append {-100} \n {100} \n {-100} \n {100} \n ;
    }
    append {1} \n {1} \n {1} \n ;
    append {ticks} \n ;
    append {30.0} \n \n ;
    append {BUTTON} \n {10} \n {10} \n {80} \n {40} \n {setup} \n {setup} \n ;
    append {NIL} \n {1} \n {T} \n {OBSERVER} \n {NIL} \n {NIL} \n {NIL} \n {NIL} \n {1} \n \n ;
```

```
    append {BUTTON} \n {90} \n {10} \n {160} \n {40} \n {go} \n {go}
      \n ;
    append {T} \n {1} \n {T} \n {OBSERVER} \n {NIL} \n {NIL} \n {NIL}
       \n {NIL} \n {0} \n \n ;
    append {BUTTON} \n {170} \n {10} \n {240} \n {40} \n {stop} \n {
      stopping} \n ;
    append {NIL} \n {1} \n {T} \n {OBSERVER} \n {NIL} \n {NIL} \n {
      NIL} \n {NIL} \n {1} \n \n ;

    // generatingsliders
    int sliderX = 55;
    int width = 35;
    foreach entity in node.entitiesAndVariables.entities {
      if (entity.initialisationNumberOfEntities.isInstanceOf(Slider))
          {
        string x1string = "" + sliderX;
        string x2string = "" + (sliderX + width);
        append {SLIDER} \n {10} \n ${x1string} \n {240} \n ${x2string
            } \n ;
        append ${entity.initialisationNumberOfEntities} ;
        append {1} \n {NIL} \n {HORIZONTAL} \n \n ;
        sliderX += 10 + width;
      }
    }

    foreach udaInit in node.descendants<concept = AttributeInit> {
      if (udaInit.initialisation.isInstanceOf(Slider)) {
        string x1String = "" + sliderX;
        string x2String = "" + (sliderX + width);
        append {SLIDER} \n {10} \n ${x1String} \n {240} \n ${x2String
            } \n ;
        append ${udaInit.initialisation} ;
        append {1} \n {NIL} \n {HORIZONTAL} \n \n ;
        sliderX += 10 + width;
      }
    }

    foreach envEntity in node.entitiesAndVariables.
      environmentEntities {
      ifInstanceOf (envEntity.initialisationLocation is Density
          density) {
        ifInstanceOf (density.percentage is Slider slider) {
          string x1String = "" + sliderX;
          string x2String = "" + (sliderX + width);
          append {SLIDER} \n {10} \n ${x1String} \n {240} \n ${
              x2String} \n ;
          append ${slider} ;
```

```
          append {1} \n {NIL} \n {HORIZONTAL} \n \n ;
          sliderX += 10 + width;
        }
      }
    }

    if (node.experiments.dataPresentation.isNotEmpty) {
      int worldsize = node.experiments.defaultWorld.pixelSize * (2 * node.experiments.defaultWorld.worldSize + 1);
      append {PLOT} \n {250} \n ${"" + (worldsize + 28)} \n ${"" + (worldsize + 258)} \n ${"" + (worldsize + 200)} \n {Model Plot} \n ;
      int finalTime = 10;
      foreach t in node.processAndScheduling.descendants<concept = TimedCondition> {
        if (t.value.isInstanceOf(NumberConstant)) {
          int newValue = Integer.parseInt(t.value:NumberConstant.value);
          if (newValue > finalTime) {
            finalTime = 10 + newValue;
          }
        }
      }
      append {time} \n {NIL} \n {0} \n ${"" + finalTime} \n {0} \n {10} \n {true} \n {true} \n ;
      append {"" ""} \n {PENS} \n ;
      string[] colours = {"-16777216", "-7500403", "-2674135", "-10899396", "-6459832"};
      int idx = 0;
      foreach c in node.experiments.dataPresentation {
        append {"} ${c.itemName} {" 1.0 0 } ${colours[idx]} { true "" "plot } ;
        if (!c.scaling.equals("1")) {
          append ${c.scaling} { * } ;
        }
        string methodName = c.method.name;
        if (c.method.is(TODO_REMOVE_Value)) {
          methodName = "";
        }
        append ${methodName} { } ${c.TODO_value} {"} \n ;
        if (++idx >= colours.length) {
          idx = 0;
        }
      }
      append \n ;

    }
```

```
    append {@#$#@#$#@} \n ;
    append {## NetLogo info tab} \n ${node.description} \n ;
    append \n {## Model Role} \n ;
    append ${node.modelDescription.role.presentation} \n ;
    append \n {## Target Audience} \n ;
    append $list{node.modelDescription.targetGroup} ;
    append \n {## Purpose} \n ;
    append $list{node.modelDescription.purposeDescription} ;
    append \n {## Research Question} \n ;
    append $list{node.modelDescription.researchQuestion} ;

    append \n {## Entities} \n ;
    boolean comma;
    foreach e in node.entitiesAndVariables.entities {
      comma = false;
      append {### entity } ${e.name} \n {with attributes } ;
      foreach a in node.entitiesAndVariables.entityAttributes {
        if (comma) {
          append {, } ;
        }
        comma = true;
        append ${a.name} ;
      }
      foreach a in e.userDefinedAttributes {
        if (comma) {
          append {, } ;
        }
        comma = true;
        append ${a.name} ;
      }
      append \n ;
    }
    if (node.entitiesAndVariables.environmentAttributes.isNotEmpty) {
      append {### environment entities } ;
      comma = false;
      foreach ee in node.entitiesAndVariables.environmentAttributes {
        if (comma) {
          append { and } ;
        }
        comma = true;
        append ${ee.name} ;
      }
      append \n { with attributes } ;
      comma = false;
      foreach a in node.entitiesAndVariables.environmentAttributes {
        if (comma) {
          append {, } ;
```

```
    }
    comma = true;
    append ${a.name} ;
  }
  append \n ;
}
if (node.entitiesAndVariables.networks.isNotEmpty) {
  comma = false;
  foreach n in node.entitiesAndVariables.networks {
    append {### network } ${n.name} \n { with attributes } ;
    foreach a in n.userDefinedAttributes {
      if (comma) {
        append {, } ;
      }
      comma = true;
      append ${a.name} ;
    }
    append \n ;
  }
}

append \n {## Model Parameters} \n ;
foreach p in node.entitiesAndVariables.modelParameters {
  append ${p.name} \n ;
}

append \n {## Attributes that Change During the Simulation} \n ;
foreach a in node.descendants<concept = UserDefinedAttribute> {
  if (node.descendants<concept = AssignAttribute>.where({~it => it.lhs.attribute == a; }).isNotEmpty) {
    append ${a.name} \n ;
  }
}

append \n {## User Experimentation Options} \n ;
append {The user can change the following attributes to create different experiments:} \n ;
foreach s in node.descendants<concept = Slider> {
  append ${s.generatedName} \n ;
}

append \n {## Data Shown in the Simulation} \n ;
foreach dp in node.experiments.dataPresentation {
  append ${dp.itemName} { } ${dp.TODO_value} \n ;
}

foreach desc in node.modelDescription.generalDescriptions {
```

```
        append \n {## } ${desc.name} \n ;
        append $list{desc.descriptionText} ;
      }

    append \n {@#$#@#$#@} \n ;
    append {@#$#@#$#@} \n ;
    append {NetLogo 6.1.1} \n ;
    append {@#$#@#$#@} \n ;
    append {@#$#@#$#@} \n ;
    append {@#$#@#$#@} \n ;
    if (node.experiments != null) {
      append ${node.experiments} ;
    }
    append {@#$#@#$#@} \n ;
    append {@#$#@#$#@} \n ;
    append {@#$#@#$#@} \n ;
    append {@#$#@#$#@} \n ;
  }
}
                                        file path : <model/qualified/
                                          name>
                                        extension : (node)->string {
  "nlogo";
}
                                        encoding : utf-8
                                        text layout : <no layout>
                                        context objects : << ... >>
```

```
base text gen component ODDGenerationSupport extends <no baseTextGen>
    {
  operation csvgen(boolean isLink, string theName, sequence<node<
      AttributeAccess>> uda, sequence<node<TODO_LetCollect>> syn) {
  with indent {
    append \n {to-report } csvgenname theName, uda {; save attributes
        to csv} \n ;
    append indent {report csv:to-row} \n ;
    if (isLink) {
      append indent {fput [" partner 1" "partner 2"} ;
    } else {
      append indent {fput ["who"} ;
    }
    foreach a in uda {
      append { "} ${a.attribute.name} {"} ;
    }
    foreach s in syn {
      append { "} ${s.attribute.name} {"} ;
    }
```

```
      append {]} \n ;
      if (isLink) {
        append indent {[ (list [ who ] of end1 [ who ] of end2} ;
      } else {
        append indent {[ (list who} ;
      }
      foreach a in uda {
        append { } ${a.attribute.name} ;
      }
      foreach s in syn {
        append { } ${s.attribute.name} ;
      }
      append {) ] of } ${theName} {s} \n ;
    }
    append {end} \n ;
  }

    << ... >>

    << ... >>
  }

  operation csvgenname(string theName, sequence<node<AttributeAccess>> uda) {
    with indent {
      string theKind = uda.size == 1 ? uda.first.attribute.name : "attributes";
      append ${theName} {-} ${theKind} {-csv} ;
    }
  }

  operation csvgencalls(node<TODO_DataCollection> dataCollectNode) {
    node<ODD> theRoot = dataCollectNode.containingRoot:ODD;
    sequence<node<Entity>> entities = theRoot.entitiesAndVariables.entities;
    string generalEntityName = entities.size == 1 ? entities.first.name : "turtle";

    sequence<node<AttributeAccess>> networkOtherAttributes = dataCollectNode.REMOVE_collect.where({~it => it.attribute.isNetworkAttribute() && !((node<UserDefinedAttribute>) (it.attribute)).type.isInstanceOf(Array); });
    if (networkOtherAttributes.size != 0) {
      append indent {<metric>} ;
      append csvgenname networkOtherAttributes.first.attribute.parent:Network.name, networkOtherAttributes ;
      append {</metric>} \n ;
```

```
    }

    sequence<node<AttributeAccess>> agentArrayAttributes =
        dataCollectNode.REMOVE_collect.where({~it => !it.attribute.
        isNetworkAttribute() && ((node<UserDefinedAttribute>) (it.
        attribute)).type.isInstanceOf(Array); });
    if (agentArrayAttributes.size != 0) {
      foreach c in agentArrayAttributes {
        string attributeName = c.attribute.isInRole(Entity :
            userDefinedAttributes) ? c.attribute.parent:Entity.name :
            generalEntityName;
        append indent {<metric>} ;
        append csvgenname attributeName, new arraylist<node<
            AttributeAccess>>{c} ;
        append {</metric>} \n ;
      }
    }

    sequence<node<AttributeAccess>> agentOtherAttributes =
        dataCollectNode.REMOVE_collect.where({~it => !it.attribute.
        isNetworkAttribute() && !((node<UserDefinedAttribute>) (it.
        attribute)).type.isInstanceOf(Array); });
    if (agentOtherAttributes.size != 0) {
      foreach entity in entities {
        sequence<node<AttributeAccess>> myAttributes =
            agentOtherAttributes.where({~it => it.attribute.parent ==
            entity || it.attribute.parent.isInstanceOf(
            EntitiesStateVariablesAndScales); });
        append indent {<metric>} ;
        append csvgenname entity.name, myAttributes ;
        append {</metric>} \n ;
      }
    }
  }
```

```
  text gen component for concept ProcessOverviewAndScheduling {
    (node)->void {
      append $list{node.codes} ;

      append \n {to go} \n ;
      with indent {
        // endbutton
        append indent {if __INTERNAL__stop [ ; stop button was pressed\
            n} ;
        with indent {
          append indent {set __INTERNAL__stop false\n} ;
          append indent {stop\n} ;
```

```
      }
      append indent {]\n} ;

      foreach e in node.endConditions {
        append ${e} ;
      }
      foreach proc in node.schedule {
        append ${proc} ;
      }
      if (node.parent:ODD.entitiesAndVariables.syntheticAttributes.
          isNotEmpty) {
        append indent {calculate ; update synthetic attributes} \n ;
      }
      if (node.parent:ODD.entitiesAndVariables.
          syntheticModelAttributes.isNotEmpty) {
        append indent {calculate-globals ; update synthetic model
            attributes} \n ;
      }
      append indent {tick} \n ;
    }
    append {end} \n \n ;

  }
}
```