Modeling Norms for Social Simulations

Increasing realism in social simulations to support decision makers in their decision making

Christian Kammler

DOCTORAL THESIS
DEPARTMENT OF COMPUTING SCIENCE
UMEÅ UNIVERSITY
SWEDEN

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

ckammler@cs.umu.se

ISBN 978-91-8070-601-8 (print)
ISBN 978-91-8070-602-5 (digital)
ISSN 0348-0542
UMINF 25.02

Cover Image by Chontida Kammler and Microsoft Copilot

Printed by UmU Print Service, Umeå University, 2025

Abstract

It is very challenging for policymakers and other decision makers to make any kind of decision on a new policy, as the reaction of a person to that policy (policy as one form of a norm) in a given situation is highly individual and based on their own subjective perspective. This becomes even more challenging in environments with a high degree of uncertainty (as is usually the case for policymakers).

Social simulations are a powerful tool for policymakers and other decision makers to support them in their decision-making process. They can try out different policies, observe, and analyze their potential effects in silico with an artificial population prior to their implementation in the real world. To build agent-based social simulations that provide this support two main challenges exist: norm (policy) realistic behavior and the usability of the simulation.

Norm realistic behavior includes differentiated norm engagement as well as seeing norms as more than just restrictions on behavior. These two crucial aspects of normative reasoning have not been addressed yet in existing work. Situated norm engagement means that people react differently to norms, and focus only on the parts that are relevant for them. Seeing norms as more than just restrictions on behavior means that people can also violate norms, and be motivated to circumvent norms. To address these two parts, we formalize different perspectives on norms to allow for agents to reason through their subjectively perceived consequences of the norm. Furthermore, we develop a novel agent deliberation architecture, called the Perspective-Based Agent Deliberation Architecture (PBADA) that is capable of representing different perspectives on norms. This allows agents to obey the norm, be motivated by the norm (obey and circumvent the norm), as well as violate the norm. Another key element of our agent deliberation architecture is that norms are explicit objects.

Having norms as explicit objects is crucial for addressing the challenge of usability of the simulation. It allows policymakers to modify them interactively in the simulation. In general, we see usability as empowering the policy maker to use the simulation in a - for them - meaningful way. Policymakers need to understand how a norm (policy) is influencing the behavior of the agents and in what way. Furthermore, policymakers need to be able to modify existing norms and add new ones on the fly. This requires interaction tools and visualization capabilities necessary to support them in this process. To address this challenge, we present preliminary work on such an interaction tool by highlighting the functionality that is necessary as well as highlighting the challenges that policymakers and other non-expert users face.

Sammanfattning

Det är mycket utmanande för politiker och andra beslutsfattare att fatta beslut om en ny policy, eftersom en persons reaktion på den policyn (policyn som en form av norm) i en given situation är högst individuell och baserad på deras egna subjektiva perspektiv. Detta blir ännu mer utmanande i miljöer med hög grad av osäkerhet (vilket vanligtvis är fallet för beslutsfattare).

Sociala simuleringar är ett kraftfullt verktyg för politiker och andra beslutsfattare för att stödja dem i deras beslutsfattande process. De kan testa olika policyer, observera och analysera deras potentiella effekter i silico med en artificiell population innan de implementeras i verkligheten. För att bygga agentbaserade sociala simuleringar som ger detta stöd finns två huvudsakliga utmaningar: realistiskt normbeteende och användbarheten av simuleringen.

Realistiskt normbeteende omfatter differentierat normengagemang samt att se normer som mer än bara begränsningar av beteende. Dessa två viktiga aspekter av normativ samspel har ännu inte addresserad i befintliga studier. Situationsbaserat norm engagemang innebär att människor reagerar olika på normer och fokuserar endast på de delar som är relevanta för dem. Att se normer som mer än bara begränsningar av beteende innebär att människor också kan bryta mot normer och vara motiverade att kringgå normer. För att hantera dessa två delar formaliserar vi olika perspektiv av normer för att göra det möjligt för agenter att resonera kring de subjektivt uppfattade konsekvenserna av normen. Vidare utvecklar vi en ny agentresoneringsarkitektur, kallad Perspective-Based Agent Deliberation Architecture (PBADA), som kan representera olika perspektiver av normer. Detta gör det möjligt för agenter att följa normen, vara motiverade av normen (följa och kringgå normen), samt bryta mot normen. Ett annat nyckelelement i vår agentresoneringsarkitektur är att normer är explicita objekt.

Att ha normer som explicita objekt är avgörande för att hantera utmaningen med simuleringarnas användbarhet. Det gör det möjligt för politiker att interaktivt ändra dem i simuleringen. Generellt ser vi användbarhet som ge politikern möjlighet att använda simuleringen på ett för dem meningsfullt sätt. Beslutsfattare måste förstå hur en norm (policy) påverkar agenternas beteende och på vilket sätt. Dessutom måste beslutsfattare kunna ändra befintliga normer och lägga till nya direkt. Detta kräver interaktionsverktyg och visualiseringsmöjligheter som är nödvändiga för att stödja dem i denna process. För att hantera denna utmaning presenterar vi preliminärt arbete med ett sådant interaktionsverktyg genom att lyfta fram funktionalitet som är nödvändig och de utmaningar som beslutsfattare och andra icke-expertanvändare står inför.

Acknowledgments

Doing and PhD and writing a subsequent thesis is a long process that requires a lot of effort. As such it is not a one person endeavor. It requires a team, and a very strong support system. Of which both I fortunately had. This is why I want to take the time now to thank the people which helped me and supported me a long the way.

First of all, I would like to thank my supervisors Frank Dignum and Nanda Wijermans who gave me the chance to pursue my research and PhD in the first place. Their feedback on my research, and their guidance was invaluable. Furthermore, it is very important to point out that they ensured mental support as well, and not only research topic related support. Especially the mental support was crucial for allowing me to complete this work, as it ensured that I could be in the right mental state to do my research.

I further would like to thank the Socially Aware AI group that I was apart of for their friendship. This includes Eren Yildiz, Maarten Jensen, Ree Mellema, and Păstrăv Pastrav. Their ongoing discussions and collaborations and various stages of my PhD helped me to further shape and narrow my thoughts, as well as broadened my vision of the field of social simulations and socially aware AI.

Further, I would like to thank Loïs Vanhée for his friendship willingness to help and support me with his advice and feedback during various points in my research process. This helped to further broaden my knowledge, as well as benefit from his vast knowledge in the field. His encouraging feedback helped me to stay positive in my journey.

Here I also would also like to thank Virginia Dignum who initiated together with Frank Dignum various collaborations and research trips four our two research groups, the Socially Aware AI group and the Responsible AI group. This helped me further explore the edges of my research and avoid becoming too narrowly focused in one direction.

When talking about collaborations, I also would like to thank the members of our ASSOCC project. Having such a big project at the beginning of my PhD was really encouraging and showed the potential of social simulations in terms of the impact they can make, especially in a crisis situation, but also the challenges when trying to communicate and collaborate with policymakers and other decision makers. The experiences I gained in this project in this multinational team which consisted of PhD students like me as well as experienced researchers provided a wealth and depth of experience. It is like Frank Dignum put it in one our our early meetings: It is like being thrown into an ocean and my supervisors make sure that I do not drown in it. So I want to thank them for not letting me drown, and Frank Dignum for the opportunity to participate in this project.

Also, I would like to thank the WASPH-HS graduate school. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program – Humanities and Society (WASP-HS) funded by the Marianne and Marcus Wallenberg Foundation.

Then, I also would like to thank my parents, Kerstin and Andreas Kammler, who encouraged me and supported me in pursuing a PhD. This then also enabled me to move to Umeå to pursue my PhD and conduct my research.

Finally, the most important person, whom I dedicate this work to, my wife, Chontida Kammler. You gave me the courage, support, and energy that I needed to complete this work. You enhanced my resilience, and made sure that I could focus on my research at all times. Your unwavering support is what made me complete and finish my research and this thesis. I am forever grateful and indebted to your unwavering support and strength.

Preface

The content and results presented in this thesis are based on and partially published in the following published papers:

Paper I — Kammler, Christian, Dignum, Frank, Wijermans, Nanda, and Lindgren, Helena. "Changing Perspectives: Adaptable Interpretations of Norms for Agents". In: *Multi-Agent Based Simulation XXII*. Ed. by Van Dam, Koen H. and Verstaevel, Nicolas. Cham: Springer, 2022, pp. 139–152.

Paper II — Kammler, Christian, Mellema, René, and Dignum, Frank. "Agents Dealing with Norms and Regulations". In: *Multi-Agent-Based Simulation XXIII*. Ed. by Lorig, Fabian and Norling, Emma. Cham: Springer International Publishing, 2023, pp. 134–146.

Paper III — Kammler, Christian, Dignum, Frank, and Wijermans, Nanda. "Utilizing the Full Potential of Norms for the Agent's Decision Process". In: *Advances in Social Simulation*. Ed. by Squazzoni, Flaminio. Cham: Springer Nature Switzerland, 2023, pp. 193–205.

Paper IV — Kammler, Christian, Dignum, Frank, and Wijermans, Nanda. "Towards a Social Simulation Interaction Tool for Policy Makers—A New Research Agenda to Enable Usage of More Complex Social Simulations". In: *Advances in Social Simulation*. Ed. by Elsenbroich, Corinna, and Verhagen, Harko. Cham: Springer Nature Switzerland, 2024, pp. 163–176.

Furthermore, the work presented in this thesis is influenced by the ASSOCC project [37] of which I was a part of during my PhD research. In this project, I specifically focused on the need-based decision-making of the agents, contributed to the developed Unity interface, ran and supported the run of various scenarios, as well as contributed to the validation of our ASSOCC model. This resulted in the first authorship and co-authorship of the following book chapters of our ASSOCC book [37] (introduction and conclusion excluded, numbers of the chapters are taken from the ASSOCC book [37]) and paper:

Chapter 3 — Jensen, Maarten, and Vanhée, Loïs, and Kammler, Christian. "Social Simulations for Crises: From Theories to Implementation". In: *Social*

Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 39-84.

Chapter 6 Kammler, Christian, and Mellema, René. “Testing and Adaptive Testing During the COVID-19 Crisis”. In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 139-166.

Chapter 12 Lorig, Fabian, and Jensen, Maarten, and Kammler, Christian, and Davidsson, Paul, and, Verhagen, Harko. “Comparative Validation of Simulation Models for the COVID-19 Crisis”. In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 331-352.

Chapter 15 Dignum, Frank, and Jensen, Maarten, and Kammler, Christian, and Melchior, Alexander, and van den Hurk, Mijke. “Challenges and Issues for Social Simulations for Crises”. In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 409-426.

Paper Kreulen, Kurt, and de Bruin, Bart, and Ghorbani, Amineh, and Mellema, René, and Kammler, Christian, and Vanhée, Lois, and Dignum, Virginia, and Dignum, Frank. “How Culture Influences the Management of a Pandemic: A Simulation of the COVID-19 Crisis”. *JASSS: Journal of Artificial Societies and Social Simulation*. 25(3), 2022.

Contents

List of Figures

List of Tables

Acronyms

ABSS Agent-Based Social Simulations

ASSOCC Agent-Based Social Simulation of the Coronavirus Crisis

PBADA Perspective-Based Agent Deliberation Architecture

ADICO Attribute, Deontic, aIm, Or else

ADICDlRO Attribute, Deontic, aIm, Deadline, Repair, Or else

V&V Verification and Validation

Chapter 1

Introduction

The recent COVID-19 pandemic is an example of the challenges that policymakers face. Policymakers, such as governments, had to act quickly to combat the spread of the virus while being in an ever evolving situation with incomplete information. What made this situation more dramatic was that people's lives were at stake and the decisions made by policymakers directly impacted them. What made this especially challenging was that the intended policies were not followed by everyone in the same manner, see e.g. Levy et al. [89]. Some people found ways around them, and some even violated them. Also, some of them did not have the intended effects. Additionally, many long-term aspects, such as the impact of school closures on learning and mental health of children is now only coming slowly to light [95]. While the COVID-19 pandemic pushed many of these decisions to the extreme, as crises do, the challenges policymakers face are not unique to them.

Another example is an intended speed limit on a street. The challenges that policymakers face in making such a decision are similar to the ones during the COVID-19 pandemic. People will react differently to the speed limit. Some may comply, others might find ways around the speed limit, such as using alternative roads, and some may ignore it. Also, some might not be impacted at all by the speed limit, if they, for example, use a bike.

These example situations show how challenging it can be for policymakers to make any kind of decision on a new policy (in any environment and for any purpose), as the decision of a person how to respond to a policy (policy as one form of a norm) in a given situation is highly individual based on their own subjective perspective. This makes it difficult for policymakers, especially in situations under time pressure, with incomplete or contradicting or ever evolving information, or pressure from outside sources (e.g. citizens, companies, institutions). All of these factors create an environment with a high degree of uncertainty that the policymakers have to make their decisions in.

1.1 Modeling human behavior for policymaker support

A way to support policymakers is to use computational models in the policymaking process [13, 56, 124]. Policymakers can use these models to gain insights in the potential effects of policies [56]. A crucial element of those models is human decision-making. Various models of human decision-making have been proposed for this purpose. An example of this is the homo economicus which assumes the individual to be perfectly rational, self-interested, and all-knowing about other agents (people), and the environment [124]. While the bounded rationality models, based on [143], acknowledge that agents have cognitive limitations and incomplete information, leading them to make satisficing rather than optimizing decisions, they still assume purely rational agents. Another example in this regard is the rational choice theory [84] which assumes a rational personal utility maximizing agent.

Although, this economic assumption of rationality is not suitable to represent (human) behavior [124, 149], there continues to be a prevalence for fully rational and economic approaches [16, 60, 161]. This is highly problematic, as they fail to capture the situated and complex decision-making with regards to norms, since people do not act rationally or always in an egoistic optimizing way.

To solve these problems agent-based social simulations (ABSS) are promising tools, as they are able to capture the complex behavior in relation to norms, as they can capture a higher degree of complexity in terms of integrated elements, as well as interactions between those elements and other agents [124]. Agents are the individuals in a simulation model that together make up the artificial population of the model [28, 145]. Having this artificial population makes models an in silico testbed for policymakers to investigate the potential effects that their policies might have, without putting anyone in harms way. An example of this potential was the call for action by [146] which called for the development and refinement of agent-based social simulations to be a valuable supportive tool to fight the COVID-19 pandemic. Another example is the recent review by [8] which reviews the use of agent-based modeling for supporting policymakers, as well as the work of [56] who report from their experiences in practice.

1.2 Challenges for using social simulations for policymaker support

To build agent-based social simulations that give this support of being an in silico test bed for policymakers, two key challenges are norm (policy) realistic behavior and the usability of the simulation.

1.2.1 Norm realistic behavior

The agents in the simulation need to show realistic human-like behavior. Crucial for this is that they need to be able to properly reason with norms (policy as one form of norms), i.e. seeing them as more than just restrictions on their behavior. This means not only that the agents can see whether the consequences of following or breaking the norm are desirable for them, but also how they interact with other parts of their reasoning process. In particular, this means that norms cannot just be seen as simple restrictions, as this is over simplified and misses all the potential behavior that can occur [82], such as their motivational aspects [20, 120] or norm violation which is an important part of norm dynamics [12, Chapter 5]. When looking at the speed limit for example, this means that the agents must be able to adapt their behavior as response to it, such as choosing a different street for driving or violating the speed limit.

This shows that norms cannot be added as a separate module on top of an agent deliberation architecture. Rather, they need to be integrated at various stages in the agent's decision-making process. Norms relate to different stages of the deliberation process, such as determining our current context (i.e. the situation we are in) or the attractiveness of each action available to us in the current situation. The decision what we (as a person) do next is based on on the interplay between our current internal state (e.g. our needs, goals), our external situation (e.g. time and place) and the norms that are active and relevant right now. Putting norms as another module on top of the deliberation process this does create an artificial disconnect and introduces rigid behavior, similarly to seeing norms as a limitation. For example, consider the situation of being at a red light at an intersection. Usually, we comply to this norm, meaning that an additional norm module would seem to work. However, there are also special situations, such as emergency situations where we might consider to violate the norm and run through the red light. These situations are then missed, as the norm is detached from the context, and might rigidly enforce norm abiding behavior.

Furthermore as shown in the speed limit example, people reason differently about the norm, based on if it is relevant for them. In terms of the speed limit example this means that people using a bike might not consider the speed limit at all, as it does not directly impact them. Other people however, that live there and use a car are potentially negatively impacted, as they now have a longer driving time. As a reaction to this they might change their behavior and leave home earlier to make up for the lost time or they might violate the speed limit. These reactions show that different people are differently impacted by the norm, and as consequence of that, the deliberation about the norm cannot simply happen in a norm module on top of an agent deliberation architecture. Rather, the norm deliberation process has to be interwoven with the overall deliberation process of the agent, and come into play at multiple stages.

With the normative reasoning aspect becoming so complex, intertwined with other aspects of decision-making, and highly subjective, the subsequent question of how complex the rest of the decision-making of the agent needs to be arises and what parts of the decision-making can be made more abstract. For example, should a social network be modeled explicitly or implicitly? Depending on the model this can be important, such as in our ASSOCC project [37] where the social network of an agent influenced their behavior via the need to do what one's network does. However, for other norms, such as the discussed speed limit, it might not be relevant, if the focus is on the individual person (represented by an agent), and they make their decision solely by themselves.

1.2.2 Usability of the simulation

Increasing agent complexity and the complexity of agent's decision-making means an overall increase of complexity of the models. Increasing the complexity of the models strongly reduces the usability of the models. In order to understand a complex model and consequently use it in their decision-making, users, such as policymakers, need to be supported to use the simulation.

Usability is of utmost importance for using social simulations in a meaningful way, especially in the context of policymaking. The strength of social simulations comes from their explanatory power to show *why* the behavior of agents is occurring in the simulation, and not only what behavior is going on. This is particularly crucial when developing social simulations for policymakers, as they are expert users with regards to the policy world but are also non-expert users when it comes to social simulations, which are typically developed with other modelers (expert users) in mind [79]. This means that they lack the plethora of background knowledge in simulations [123]. For example, policymakers must be able to see and understand in the model why agents violate the speed limit or in the case of our ASSOCC project [37] why some agents adhere to the lockdown while others are violating it. Furthermore, this information needs to be presented in such a way that the policymaker can quickly conceive the information. Otherwise, the simulation just becomes a black box.

A challenge related to this is the modifiability of the simulation. Policymakers (users) must be able to tailor the simulation towards their needs. A key factor in this is the modifiability of norms. Policymakers need to be able to modify the existing norms and add new ones on the fly [82]. This is crucial for using social simulations as an in silico test bed, as otherwise no testing of alternative policies can be done.

For example, when they observe that the speed limit does not work as intended, they need to be able to alter the speed limit from e.g. 30 km/h to 20 km/h or increase the punishment for violating it. Furthermore, they might want to add a new norm, such as banning certain kinds of traffic, e.g. cars, at certain periods of the day. Consequently, policymakers need to be supported to take actively part in the modeling process [82].

Enabling the policymaker to do so is a highly difficult task which has not been attempted yet. Furthermore, it requires the norms to be modeled, as explicit objects in the simulation. Only then can they be modified by the policymaker (user) and added on the fly. This is not possible when they are implicitly modeled through the actions of an agent.

For example, in our ASSOCC project [37] we modeled norms implicitly by integrating them into the actions of the agents by providing different need satisfaction gains depending on a norm being (de)active. Doing it this way does not allow for norms to be changed easily. In our ASSOCC model, when one wants to change a norm, they need to go through all the actions to see if they are affected by the norm and adapt them according to the norm change. This becomes even more challenging when adding a new norm. In this case it might not even be clear which actions are potentially affected, and thus, long discussions about each action have to happen to determine this.

Explicitly modeling norms solves these drawbacks, as the norm does only need to be changed or added in one place and can then be taken into account by the agents when they deliberate about their actions. For example, when adding a speed limit, one does not have to look at all actions potentially affected by it, such as driving a car or going by bus (as the bus must adhere to the speed limit). Rather, the norm is added and specified in one place and the agent is then using the norm when they decide on their action to, e.g., go by car or take the bus or bike.

This also imposes a great challenge on existing platforms, such as NetLogo [162] or Repast [117]. Policymakers are non-expert users in terms of social simulations [79], and as such they need extra support to add or modify norms. We (modelers) cannot expect them to the able to specify norms in such a detailed and formal level that they can be used by the agents in the simulation. However, currently existing platforms are tailored towards modelers (expert users) and thus, do not provide the support and functionality necessary, to be used by policymakers.

1.3 Research Questions

It is clear that these challenges, focusing on norm realism and modifiability and addition of norms, are clearly connected. Therefore, we propose the following main research question (Main RQ) that we investigate in this thesis which we will then concretize more in concrete research questions (RQs) that we will address in this thesis.

Main RQ: How to increase norm realism and usability of social simulations to support policymakers in their decision making?

This research question encapsulates the overall background of the work that will be conducted in this thesis. However, this question is also inherently abstract and can be answered in various ways while also not being concrete what is to be investigated.

To make things more concrete we translate the above abstract research question into the following four concrete research questions (RQs) that we will address in this thesis.

RQ 1: How can different interpretations of norms be formalized?

The key element for dealing with norm realism is that people focus only on the parts of the norm that are relevant for them [82]. For example, a restriction of the number of customers in a restaurant or shop in COVID time means possibly less income for the owner, while it means less opportunity for customers to shop or dine out. Each of them (customers and owners) will try to cope with this restriction in different ways. Customers may try to find times to shop or eat that are quiet or possibly reserve a spot in advance. A restaurant or shop owner can try to facilitate customers by creating a reservation system, making more space available, but might also reduce their personnel. Thus a key challenge, which has not been dealt with when supporting policymakers is to allow for this situated engagement with norms. An answer to RQ1 is exactly dealing with this gap in the research.

Answering this question requires multiple steps. The first one is to identify and formalize concepts that can be used to provide a subjective worldview for the agents to allow them to reason subjectively about norms. This is then brought together in the second step when formalizing a perspective on norms. Finally, the third step of answering this question is then also to provide a suitable formalism for norms. To enable agents to focus on the parts of the norms that are relevant and make them explicit objects, we need to provide a novel formalization of norms. We will address these parts, and thus RQ1 in Chapter 3.

RQ 2: How does an agent deliberation architecture need to look like to integrate different perspectives on norms?

Based on the answers in RQ1, this question aims at identifying and developing a novel agent deliberation architecture which is capable of dealing with different perspectives on norms. Concretely, this means to address the question of how the formalizations developed in RQ1 can be integrated in an agent deliberation architecture such that agents can use them. Policymakers need to be able to use a simulation that is capable of dealing with different perspectives on norms. It is not enough to have just formalizations of different perspectives of norms. They also have to be integrated into an agent deliberation architecture. Furthermore, this agent deliberation architecture needs to be a general purpose architecture suitable for a variety of situations, such as crises (e.g. COVID-19, or global warming) or general decision-support (e.g. a simulation of a city to understand where speed limits could be placed). Also, this agent deliberation architecture needs to use norms as explicit modifiable objects in the agent's decision-making process, so that they can be modified and added on the fly.

To answer research question RQ2, we will develop a novel agent deliberation architecture, called the Perspective-Based Agent Deliberation Architecture (PBADA) in Chapter 3 and discuss why we made the PBADA architecture the way we did and argue for our choices. To show that our proposed agent deliberation architecture is indeed able to deal with different perspectives on norms, and is showing norm realistic behavior, i.e. obeying a norm, violating it or obeying the norm and circumventing it, we instantiate our PBADA architecture in a model in Chapter 4 to verify and validate our proposed PBADA architecture.

RQ 3: How can users be supported to change norms and add new ones during the simulation run?

This research question addresses the second part of the main research question, namely increasing the usability of simulations. Policymakers must be able to change parameters of existing norms, such as their associated punishment, or also add new norms during the run of the simulation, if they identify specific situations that they want to target. For example, if the speed limit of 30 km/h does not decrease the accident rate, policymakers must be able to modify it to, e.g., a speed limit of 20 km/h or increase the fine for violating it. Furthermore, it might happen that due to the speed limit, accidents in another part of the town increase. To tackle this problem, policymakers might want to add another norm targeting that area. Policymakers must be supported in these explorations by the simulation.

However, changing elements in the simulation is not trivial and the consequences of what a change means are often very unclear. The user interaction needs to be such that users can get explanations, and be able to trace what is happening to an agent. Furthermore, they need to able to find out how the agents are interacting with the existing norms. Only then are they able to fully understand what is going on and what potential changes need to be made.

In Chapter 5 we will show some preliminary work on how non-expert users (such as policymakers) of complex social simulations can be supported to understand what the simulation is doing and also what are the consequences when norms are added or changed.

RQ 4: What support do non-expert users need from a user interaction tool to be empowered to use the simulation in a for them meaningful way?

Connected to the previous research question is this research question. Policymakers can only make decisions based on testing various norms (policies) and use the results of the simulation, when they know and understand what is going on in the simulation. As such, the simulation needs to provide the understanding and exploratory functionalities required by the policymaker to understand the simulation and use it in a, for them, meaningful way. This is to allow the policymaker to investigate and explore the simulation from various angles and on various levels of detail and overview.

RQ3 and RQ4 are strongly connected, as they both are the pre-requisites and consequences of each other. A policymaker can only modify a norm in their desired way, if they understand the potential consequences and know where they have to start and what the current situation in the simulation is. Furthermore, for a policymaker to use the simulation in a meaningful way, they must be able to modify the norms as well to see potential outcomes and behavioral impacts.

Consequently, we will answer RQ4 togther with RQ3 in Chapter 5. The preliminary results presented in this chapter also aims at exploring the simulation and getting meaningful (relevant for the user,e.g. policymaker, of the simulation) insights.

1.3.1 Contributions

By addressing all the research questions above, we provide the following contributions with our work presented in this thesis. We aim to provide some preliminary work on how non-expert users (such as policymakers) of complex social simulations can be supported to understand what the simulation is doing and also what are the consequences when norms are added or changed. With this, we aim to bridge the gap between the policymaking and the modeling world, thereby enabling a greater adaption of agent-based social simulation by policymakers to use them as decision-support tools in their work.

This is further strengthened by our aim to contribute to norm-realistic behavior of agents by integrating different perspectives on norms. With our aim to develop a novel agent deliberation architecture, policymakers can get a better understanding how their intended norms potentially affect the different groups of people (represented by different perspectives in the simulation). This then allows them to make better decisions and tailor their planned policy towards the desired outcome by taking potential consequences into account that would otherwise, without the use of our proposed PBADA agent deliberation architecture, be missed. This is because with our PBADA agent deliberation architecture we provide a deliberation architecture in which understanding of where and at which points in the agent's deliberation process, the norms are affecting them.

While these contributions aim to improve the real world use of social simulations, our contributions also provide significant scientific advancements in the field of agent-based social simulations. By formalizing different perspectives on norms, and providing a novel agent deliberation architecture, we aim to further advance norm-realistic reasoning, and foster the development of perspective-based social simulations. This research is also contributing to the understanding of norms, and how they affect our (human) behavior.

Our preliminary work on an interaction tool contributes to further understand the support that (non-expert) users need when dealing with formalized systems, such as social simulations. In this way, our research does not only contribute to the field of social simulations specifically, but to any field where non-expert users deal with complex formal systems and try to interact with them. our research aims to provide insights into potential requirements and next steps to be taken.

1.4 Structure of the thesis

The relation to the different research questions can be found in Figure 1.1. The figure also provides a connection to the different chapters in this thesis, and where the concrete research questions are answered with some indications on what ought to be done in that chapter. The main focus is on the core parts of our thesis. Hence, the background chapter (Chapter 2), and the overall conclusion chapter (Chapter 6) are omitted in the figure.

The figure shows that our thesis contains two main deliverables: a novel agent deliberation architecture, and preliminary work on a user interaction tool.

The novel agent deliberation architecture, which we call PBADA (Perspective-Based Agent Deliberation Architecture) will be formalized and conceptualized in Chapter 3. This chapter includes what we mean by different perspectives on norms,

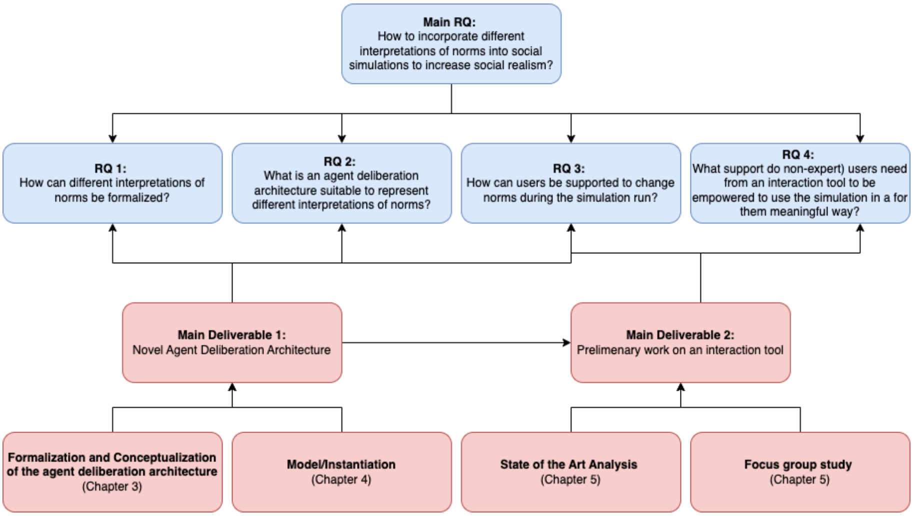


Figure 1.1: Relation of the thesis work and the research questions

i.e. what a perspective must contain. Furthermore, justifications for the formalization are provided. Since this will impose requirements on the formalization of norms, also that formalization with its corresponding justification are provided. In the next step, our PBADA is conceptualized, and explained. In the final step in Chapter 4, our architecture is instantiated in a model to verify and validate our PBADA agent deliberation architecture

Our preliminary work on an interaction tool for non-expert users (such as policy-makers) will be presented in Chapter 5. In this chapter, we analyze the requirements for a novel user interaction tool, provide solutions, and propose a new research agenda. The agent deliberation architecture also slightly touches on the research questions regarding norm modification The structure of the conceptual architecture and the norm formalization are imposing requirements of the support necessary for the user of the simulation (policymaker). The user needs to be able to modify the elements that are part of the formalization of the norm. Furthermore, the formalization of a perspective on norms also impacts the way the user can explore the simulation and modify it.

Chapter 2

Background & Related Work

2.1 Introduction

Social simulations can be a powerful tool to support policy makers in their decision making by providing an in silico test bed to conduct what-if analyses. Policymakers can investigate the potential effects of their policies before they implement them in the real world. In other words, instead of conducting experiments with the target system (real world) itself, a model is developed which describes the relevant parts of the target system in an abstract and simplified way. Figure 2.1 shows the process of model building and simulation adopted from Gilbert & Troitzsch [55].

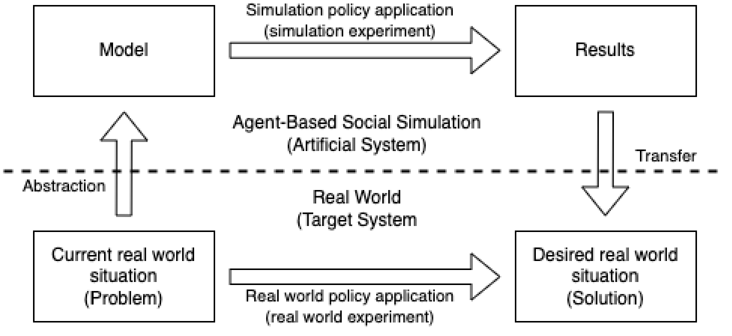


Figure 2.1: Process of model building and simulation adopted from Gilbert & Troitzsch [55], adopted with terminology relevant for policymaking for our thesis

As not all mechanisms of the underlying system, i.e. the real world, can be mapped 1:1 in a model, assumptions must be made to formalize the system's behavior. This is a highly complex question, because models or even more so our intended agent deliberation architecture, must be able to accommodate a variety of scenarios and possibilities for policymakers to play with the simulation, alter it, and use it, in a for them, meaningful way. To allow for a variety of scenarios, different concepts need to be integrated in the model which makes the model itself more complex. Furthermore, policymakers need to be able to instantiate the model in different ways. This also makes model building for policymaking different from developing simulations of a specific situation or use case. Simpler models are often limited with respect to how they can be parameterized and thus, do not provide the same variety of scenarios that can be investigated.

To tackle this question, it is important for us to understand the role that norms play in human behavior and how legal norms (policies) fit in. This is what we do in Section 2.2. Furthermore, we are not the first ones that are dealing with norms and agent-based models. There is a plethora of work on agent deliberation architectures and agent-based models dealing with norms which we will investigate in Section 2.3.

Policymakers need to be able to trust the agent deliberation architecture that we develop in this thesis. Therefore, it is important that we understand how to verify and validate our proposed agent deliberation architecture. For this reason we look at different approaches for verification and validation of social simulation architectures and models in Section 2.4. Finally, policymakers need to be able to actively interact with the simulation in a, for them. meaningful way. Therefore, we need to look at existing approaches for user interaction tools in Section 2.5.

2.2 Norms, Values, and Motives

2.2.1 On Norms and their formalizations

Norms have been studied for a long time by various disciplines, for example in philosophy and sociology [9, 12, 62], psychology [21], biology, economics, and law [26]. This alone however does to make it useful for the use in agents in simulations, as those definitions and discussions only exist in plane english but not in a formalized way.

A way to formalize norms is the use of deontic logic [156]. Deontic logic is the logic that deals with obligations, permissions, prohibitions, and related concepts. Furthermore, it deals with actual and ideal behavior [159]. Obligations are must do actions, or must norms. For example, if someone is obliged to do action a, then they must do action a. Likewise, if someone is prohibited to do action a, then they must not do action a. This means prohibitions are must not not norms, meaning that something is forbidden. A permission is a weaker version of an obligation, as it refers to a can action, or can norm. This means that something can be done but not must be done. For example, if someone is permitted to do action a then they can do action a but do not have to, and also cannot do action a. In general, the works done in the Deontic Logic in Computer Science (DEON) community, see e.g. [19, 57, 77, 108, 110], provides great advancements in the use of deontic logic for computer science. From these works, two main views on norms emerged: the implicit view on norms, and the explicit view on norms.

In the implicit view norms, see e.g. Conte & Castelfranchi [23], norms are seen from patterns of behavior. This means that based on a given behavioral pattern, normative behavior is deduced and interpreted. A major drawback of this approach however, is that it is not clear or explicit how and where norms connect to the different parts of the reasoning of the agents. Thus, the agent remains in a way a blackbox. This is not suitable for our research.

The explicit view on norms connects the norms to the different parts of the reasoning of the agents. In Dignum [33] and subsequently Dignum [34], for example, attempts have been made to transform norms into obligations which can then be used as goals by the agents [20]. Other influential work on connecting deontic logic and agent reasoning has been done by Meyer and collaborators, see e.g. [105, 106, 107, 109, 108, 159]. In [159] for example, Meyer & Wieringa provide an overview of the applications of deontic logic in computer science [159]. Furthermore, in [107], Meyer integrates dynamic aspects in deontic logic. Furthermore example work of Meyer and collaborators can be found in [109] for their work on deontic logic and normative system specifications or in [105] for epistemic logic for AI.

To make norms more practical for policy design and analysis, Crawford and Ostrom introduced the ADICO framework [25], also known as the institutional grammar. The grammar provides a general format for rules [144], and was specifically developed to analyze institutional statements [25], such as (legal) norms. The goal is to understand action situations, and "provide a detailed depiction of what actions are allowed, permitted, and forbidden under specified conditions and often with specific sanctions for actors" [142, p.79]. Thus, it offers a way to identify the core elements a policy is comprised of [142]. This specified way is very important for our case, as policies (laws) need to be very precisely formulated so that they can be implemented in social simulations. Siddiki et al. [142] go as far to say that the purpose institutional grammar (ADICO grammar) is "[...] analogous to genetic codes in living cells [...]" [142, p.81]. Furthermore, this grammar has been used in a variety of simulations, see e.g. [53, 144] adding to its validity and applicability for agent-based social simulations. The ADICO grammar consists of the following five elements [25, 82].

A Attribute: the person(s) responsible for adhering to the (in our case) norm

D Deontic: The deontic element (obligation, prohibition, permission)

I aIm: The targeted action of the norm, e.g. the action which is forbidden

C Condition: The circumstance under which the action is e.g. forbidden

O Or else: The punishment for violating the norm

However, not all elements need to be present. Crawford and Ostrom distinguish three different types [25]: shared strategies, norms, and rules. Shared strategies only have the AIC components, norms have the ADIC components, and rules have all ADICO components. Relating this back to policies, it can be seen that rules with their punishment (or else) component, closely resemble the notion of policies, as they also have a punishment for violating them. The notion of norms from Crawford and Ostrom and Ostrom can be related to social norms. While social norms usually also come with a form of punishment, the authors omit that, as they assume adherence to them due to societal pressure [25].

Further extensions of the grammar have been proposed. An important extension is the split of the aIm (I) component into a verb and an object. This has been proposed separately by [144] and [142]. Izquierdo & Smaigl call it *I-V* (verb) and *I-O* [144]. Siddiki et al. [142] call the object *oBject (B)* while the verb remains the *I* from the traditional grammar [142]. This split allows for example for more flexible reasoning and combinations of verbs and objects by agents. [144]. Further extensions of the grammar can be found in [47, 48].

Various kinds of norms have been used in agent systems over time. Mellema et al. [101, 102] provide the following distinction between different norms that are used in agent systems: social norms, legal norms, moral norms, and constitutive norms.

Social norms describe the expected behavior when being part of a group. Deviating from that behavior can lead to sanction, and punishments from the group [102]. Example social norms can be forms of greetings like handshakes or specific behavior in specific situations. Moral norms are different from social norms, as they are more individual and not tied to a specific behavior. Moral norms focus on what people see as ‘the right thing to do’ and what aligns with their values the most [101, 102]. Legal norms differ from the other two types as they are neither emerging from a social group, nor from the values of an individual or a group. Legal norms are rules declared from an institution and enforced by an authority and no the society in itself [102]. Constitutive norms are used to classify elements [10]. They can be seen as counts-as rules that classify behavior and objects [101]. For example, that an electric bike counts as a bike, or that buying something online via a buy button counts as excepting the terms and conditions of the specific shop [101].

The main relevant difference between the various kinds of norms, besides their definitions, is how the violation is handled [101, 102]. This makes a key difference how the agents reason about it. Those punishments can be fines, e.g. when violating legal norms, can make other actions forbidden or urgent, or can result in reactions of other agents, e.g. when violating social norms. These punishments are not exclusive to the different types of norms, but can also be shared. But the main two elements which connect to the deliberation process of the agent are how the violations and punishments are modeled, and how the the norm monitoring mechanism works. This enables for potential reasoning through the consequences of the norms and what a norm means for a person.

2.2.2 Norms influencing behavior

Policymakers have an intention and a goal of what they want to achieve in a given situation, e.g. reducing the spread of COVID-19. But the only way they can influence the behavior of the people is by using legal norms (policies). They cannot influence the social norms governing a restaurant visit for example. Nonetheless, they have to be aware of them to understand potential requirements on the intended policies. Existing (social) norms can influence the likelihood of adhering to the intended polices, as argued in the compliance model [124, 127]. Within this framework it is argued that the likelihood of people adhering to the (new) legal norm depends on the fit or alignment with their already existing norms [124, 127][1]. Similarly the theory of planned behavior [2] argues that existing social norms influence legal norm conforming behavior [2, 124]. If the existing social norms are supporting the (new) legal norm or at least do not support norm violations, then the likelihood of adhering to the (new) legal norm is high. Vice versa, this also means that there can be a conflict between existing social norms and the (new) legal norm, if the existing social norms are contradicting the (new) legal norm.

Norms also influence other social rules that govern our behavior and can be used to govern the behavior of agents in a simulation. While legal norms are the norms that can be controlled (created or removed) by the policymaker, they do not exist by themselves in a vacuum [124]. They exist in a web of social rules [124], such as conventions, social practices, habits, and other norms [102] (as already touched upon in the previous section).

The relation between the social rules can be found in Figure 2.2. The figure shows the social rules discussed by Mellema et al. [101] and how they can emerge from each other. The black arrows show the path of emergence between the social rules, meaning how one social rule can transform into another social rule. The figure was adapted by adding the following black arrows to make it explicit in the figure, based on the description in their paper.

Social norms can be institutionalized and emerge as legal norms, or come to be seen as the 'right thing to do' and thus evolve into moral norms [101] (social to moral norm black arrow already existing, so we added only the black solid arrow to the legal norm). These laws and moral expectations then shape interactions with others, guiding what is acceptable and influencing social norms. In the case of legal norms this can either be a supporting influence, when a legal norm (law) aligns a social norm, or create a conflict, if a legal norm (law) is contradicting a social norm [124].

Similarly, moral norms can also be institutionalized to be come legal norms (black solid arrow added), if the authority sees active enforcement of that norm as necessary [101]. In both cases, institutionalization of moral norms as well as institutionalization of social norms, the resulting legal norm (law) tends to become more abstract to cover a wider array of situations [101].

Another interesting possibility is that legal norms can also transform into social or moral norms when an official legal regulation (law) is repealed [101] (black solid lines added). When this is the case the behavior enforced by the previously existing law is still persisting even though the law itself (that enforced it) had been repealed. As an example for this, Mellema et al. [101] used an increase of the speed limit on some highways in the Netherlands. Although, the speed limit was raised from previously 120 km/h to now 130 km/h, many drivers (especially shortly after the raise) continued to follow the previous 120 km/h speed limit [101].

Furthermore, habits were added as a box within social practices, as habits are referred to by the authors as individual, one-person social practices [101]. Habits are highly individual behavior rules [101], such as brushing one's teeth after getting

[1]Another contributor is the trust in the policy maker issuing the new legal norm [124, 127]. But this discussion is out of the scope of our thesis.

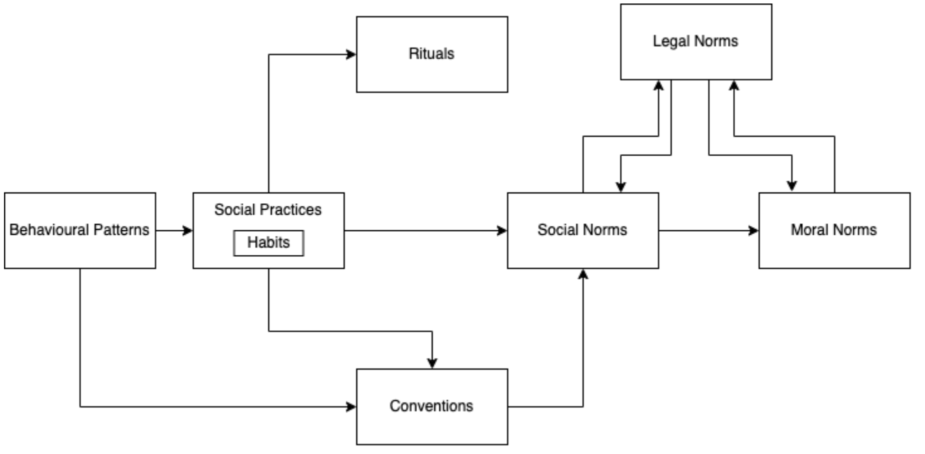


Figure 2.2: Path of emergence of social rules, adapted from Mellema et al. [101] (legal norms and habits added, based on the arguments in the author's paper). Black arrows from left to right show emergence of social rules

up in the morning, and deserve some attention in the context of this thesis[1]. In general, they can be seen as "a more or less fixed way of thinking, willing or feeling acquired through previous repetition of a mental experience" [4, p.121]. The reason that habits are brought up here is that they make up the majority of a persons individual behavior, see e.g. [22]. This means that policymakers should take the typical behavior of individuals into account, as it might be that a habit is very strong, and people are not willing to change it for a (new) legal norm. The reason behind this and why it is hard to change a behavior is that after habit formation, a disconnect between the original reason of the behavior (goal) and the action itself (the plan) is tacking place [101, 165]. According to Mercuur et al. [104], it is not enough that people are informed about negative effects of their habits, after this disconnect is made. In this sense, habits become "automatic behaviour" [101, p.7]. If a (new) legal norm now interferes with a habit, the person needs to actively deliberate about achieving the intended outcome (goal) and to respond to the current situation [101]. However, this does not mean that the person will alter their habit to adhere to the norm. It might be that the person recognizes the norm, deliberates about it, but not adhere to it, if the person deems the habit more important than the norm.

2.2.3 Values and Motives influencing normative behavior

When dealing with norms, it is also important to look at ways in which people interact with and possibly violate norms. Here values and motives come into play. Values are used when determining to follow or violate norms, and motives are personal criteria

[1]Compared to the other social rules shown in Figure 2.2 for which the discussion is out of the scope of this thesis

for determining ones behavior. This is why in this section, we will describe them in more detail.

2.2.3.1 Values

Values have been part of the psychological debate for a long time. Based on the various definitions of values, see e.g. [129, 139], three common features can be extracted [88, 138]. First, values are relatively stable beliefs about desirable end-states of reality that motivate action. Second, values serve as general guiding principles for behavior and decisions over a wide range of contexts [3]. Third, a person's values are ordered based on their relative importance to that person [138]. This means that values are evaluation criteria for events and behavior [129, 139].

A famous and widely used value system is the Schwartz Value system [138], see Figure 2.3, which has received an extensive amount of attention and empirical validation [130].

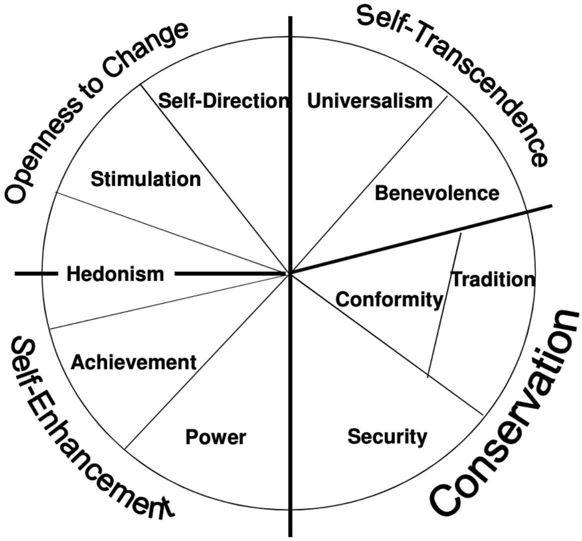


Figure 2.3: Schwartz Value System from Schwartz [138]

In the Schwartz value system [138], the values are ordered in a circumplex in such a way that values which are related are next to each other and values which are opposing each other are further away from each other. Values placed close to one another in the circumplex are considered mutualistic (tend to harmonize with one another) and values placed further away from one another become increasingly antagonistic (are generally in conflict with each other) [88, 138].

There is plenty of more work on values. Other example theories include [124]:

- Intrinsic values, such as life, health, pleasure [97, 124]

- Lived values, such as health, safety, belongningness [58, 124]
- Noneconomic/invisible losses, such as identify, life style, health dignity, education, tradition [124, 141]

The first work of formalizing values was done by van der Weide [157]. This formalization is based on the following two assumptions: values are criteria to guide behavior, and abstract values (e.g. Schwartz values [138]) should be connected to concrete states in the world [36]. To do this value systems (or value trees) can be used to connect abstract values to more concrete intermediate values, and the concrete states in the world [36]. For example: The Schwartz value of security [138] can be connected to the intermediate value of health which can be related to the world state of biking to school [36]. This means that biking to school promotes health which promotes security and thus, biking to school promotes security [36]. However, having value systems can result in debates about which world state to connect to which value [36] and in what way (positive or negative).

2.2.3.2 Motives

The theory of motives was originally introduced by McClelland [96]. McClelland describes the following motives [36, 96]:

- Achievement
- Affiliation
- Power
- Avoidance

The achievement motive drives us to progress from our current situation to something better[1]. The affiliation motive drives us to be together with other people and socialize. The power motive drive us to have the power and thus the ability to control the environment. This means a drive to be autonomous and have the ability and skills to deal with the situation at hand and perform the desired action. Finally, the avoidance motive drives us to avoid situation in which we do not know what to do or what to expect from others [36, 96].

Within this theory, McClelland [96] suggests that all humans have all four motives at all times but with varying degrees of importance. This means that humans prioritize certain motives over others. For example: a restaurant owner can be more achievement driven, and is thus seeking states in which they feel that they progress, e.g. having more money. Another example could be: a guest that has a high priority in the affiliation motive might go out a lot to socialize with their friends.

The connection between the motives and the actual actions can be described as follows. The environment provides cues in terms of affordances to signal what behavior can be performed now. In other words affordances describe the "action possibilities

[1]Better here strongly depends on the person, and can broadly be anything, e.g. personal success

provided to the actor by the environment"[1] [80]. For example: A chair provides the action possibility to sit on it. Recognizing these affordances results in some incentives for a person. These incentives can then trigger and be combined with the motives of a person to form a concrete motivation to perform an action [36]. For example, seeing a running track when walking outside can trigger the achievement motive to improve ones fitness and making the motivation to go running salient. However, not every motivation will lead to a concrete action [36]. Another crucial part in this decision-making process is the usage of the skills of a person to perform the motivated action. Furthermore, other motives, such as avoidance can play a role to decide if running is safe enough for example or putting the person at risk. Additionally, the current importance of the motive plays a role as well. For example, if a person was running yesterday, it is not likely that they might run today again, compared to seeing a friend that they haven't met for a week which triggers the affiliation motive to meet the again.

2.3 Existing Agent Deliberation Architectures

Various agent deliberation architectures have been developed over time based on the elements described in the previous sections. For example Castelfranchi et al. [20] nicely shows that norms come into play at various stages at the deliberation process. [92, 133] are examples for surveys about existing work.

Many agent deliberation architectures are either directly based on or take inspiration from the BDI model of agency [128]. The BDI model of agency is also called the Beliefs-Desires-Intentions (BDI) architecture [128] and represents a cognitive agent deliberation architecture. The beliefs of an agent represent its current understanding of the world, and themselves, i.e. what they believe to be the current situation for them. Desires represent the goals, and more motivations that the agent has, i.e. the states they wants to bring about. Intentions are representing the actions and plans that the agent has available to achieve its desires.

The key to understanding the various agent deliberation architectures that have been developed, is how they treat norms. Existing agent deliberation architectures can be divided in two sections: norms being implicitly modeled within the agents and their available actions, and norms being modeled explicitly as explicit objects in the simulation. Furthermore, various theories, such as utility functions, values, and motives have been used to connect the reasoning about norms to the internal states of the agents.

2.3.1 Implicitly Modeled Norms

Implicitly modeling norms does not mean the absence of norms. Rather, the norms are integrated into the agents and their available actions. This means that by reasoning about their actions the agents are reasoning about the norms. But crucially there is

[1] https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/affordances, accessed: 02/20/2021

no explicit reasoning about the norms themselves. An approach that has seen a lot of attention here is the one based on utility functions which we will explain next.

2.3.1.1 Utility functions

Utility functions stem from economics [36] where fundamentally utilities are ascribed to a good [72]. In general, utility functions can be seen as an ordering over sets of preferences [30, 36]. The idea behind utility functions is also that adding more of a set also increases the utility of it [36]. For example: Having 1000$ has a higher utility than having 990$. The advantage here is that a numerical representation is given for the preferences which makes it easy to use them in algorithms and the decision-making process of the agent [36]. Furthermore, they allow for a situated decision-making to some extend, as different situations can result in different decisions. For example: Adhering to norm one might have a higher utility than violating it in a given situation, and thus the agent will adhere to the norm. While adhering to norm two might have lower utility than violating that norm, and therefore the agent will violate norm two. However, it is very important to note here that deciding to adhering to norms does not mean that the agent is explicitly deliberating about the norms. Rather, the agent is looking at the actions and the utilities they provide. If the action providing the highest utility is the one that is in line with the norm, then agent adheres to the norm. If the action providing the highest utility is violating the norm, then the agent is violating the norm.

Various agent architectures have been developed incorporating utility functions, see e.g. [5, 18, 22, 103, 120, 121, 155]. EMIL-A [5, 18] and the work by Panagiotidi et al. [120, 121] also incorporate context sensitivity for agents.

In the EMIL-A [5, 18] architecture for example, the agent can decide in a given context to adopt a new norm if it is not already in their memory and comply to it. This decision can be based on the following factors: adopting another agent's goals, avoiding punishment, or gaining approval of others [65]. This decision is based on a utility function, and the action with the highest utility (norm adoption and compliance based on one of those reasons, or norm violation) is chosen. While the EMIL-I-A architecture [155] extends the EMIL-A [5, 18] with norm internalization processes, it still uses utility functions as their basis for the agent's decision-making process.

The approach by Panagiotidi et al. [120, 121] for instance, is using a total cost utility which the agent tries to minimize in its planning. Here non-normative behavior has a higher cost than normative behavior. This means that the agent can reason about the norms whether to violate them or not through the actions. Usually, a utility function is conceptualized in such a way that norm breaking behavior results in utility penalties [45].

However, it is exactly in this situated decision-making where some drawbacks of utility functions are. The main issue is that the utility and the subsequent preference orderings are fixed [36]. This means that while different situations can result in different behavior, *the same* situation will always produce *the same* outcome. Consequently, the history of the agent in terms of their behavior is not taken into account. This is crucial however, as the agents , or we as humans, determine our behavior based on our past behavior. For example: Work that is giving 100$ to a person as a reward is attractive, if the person only has 100$ in their bank account. If someone does some work which will give them 100$ is attractive if they only have 100$ in their bank account. But when the person has 1000000$ in their bank account, the same work for 100$ is not that attractive for them. In terms of the utility functions the utility of doing the work for 100$ is the same however, meaning that if on a Friday evening for example they do this work, they will always do it, regardless of how much money they have in their bank account. This situatedness of the decision-making process also influences our normative behavior. Sometimes it might be more beneficial to adhere to a norm, while in other case it might be more beneficial to violate that norm. Defining a utility function for every possibility is not feasible [78].

This becomes even more problematic in changing environments. The policymaking environment is such a changing environment. It is not static. But utility functions only work in static environments [37, 78, 80]. To provide policymakers with the possibility to modify existing norms, and add new ones to support them in their decision-making process, the utility function also needs to be modified and extended when an existing norm is modified or a new norm is added [78]. This is practically not feasible, and not desirable, especially in a dynamic environment with many norms.

2.3.1.2 Motives & Values

To move away from the challenges and static behavior of utility functions, other work has been done to incorporate more social theories, such as motives and values. This is also to tackle the challenge that the utility functions are transporting the image of a rationally deciding agent, whereas in the real world humans do not make rational decisions, in the strict economical sense [61].

One example of using values is the work by Dechesne et al. [31], and Di Tosto & Dignum [32] for motives and values who investigated the compliance of the smoking ban introduction in several European countries. Dechesne et al. [31] investigated the compliance with the smoking ban policy by using the concepts of values, based on the cultural value theory of Schwartz [137], social norms, and culture by Hofstede & Hofstede [69]. The results showed that indeed incorporating these social concepts provide insights and can be used to explain the compliance of policies (legal norms) in a specific culture. Di Tosto & Dignum [32] used a similar approach by incorporating values and drives into the agent's decision making process. They use values to prioritize between the specific drives [32]. The drives themselves represent needs that the agents want to satisfy, such as affiliation or comfort [32] which can be related to the motives by McClelland [96]. The authors showed that their model consisting of values and drives, was able to realistically handle the behavior of agents in the situation of conflicting drives, based on the individual preferences of the agents, i.e. the relative importance of certain drives compared to other drives.

A very recent work is the Agent-Based Social Simulation of the Coronavirus Crisis (ASSOCC) project [37]. In this project, the potential effects that policies have on the spread of the coronavirus and subsequent behavior of the agents, such as closing schools or lockdowns, were investigated. The motives of McClelland [96], and the Schwartz value system [138] were combined into the watertank approach from [40, 66, 65] and called needs of the agents, resulting in a homeostatic needs model [36, 75] allowing for stable behavior of the agents over time. This allowed for the investigation of potential effects of polices on the needs of the agents, and how their behavior subsequently changes such that the agents still can satisfy their needs. To understand which actions the agents can take, the concepts of affordances [54] was used to determine which behavior is most salient at a given location, e.g. the office affords to work [36]. Finally, the cultural dimensions from Hofestede [52, 70, 88, 111] were used to connect the values of agents to their ascribed culture to allow the investigation of the potential effects of culture of the spread and policies of COVID-19 [52, 88].

A major drawback of this work however is that the norms are implicitly modeled in the expected need satisfactions of an action. For example, going out to meet friends at a private leisure place would give a different need satisfaction for belonging (need to socialize with friends) given that social distancing was required [37]. As a result, when a norm is modified or new one is added, one has to go through all existing action to see if something has to change in the need satisfaction gains of that action. For example, if the social distancing radius would be increased, one has to see how this potentially affects the belonging need satisfaction gain of going to a private leisure place, and subsequently change it. Similar to utility functions, this is practically not feasible, and not desirable, especially in a dynamic environment with many norms.

There is much more research done on integrating social theories. Further examples can be found in [11, 24, 71, 150, 151, 164].

2.3.2 Explicitly Modeled Norms

For people to engage differently with norms, allow for the modification of existing norms, and addition of new norms, norms need to be modeled explicitly. This is to model them as explicit objects in the simulation. So far not much work has been done on modeling norms explicitly.

Early examples include the BOID [15] and B-DOING [39] architectures with recent work done by Heidari [65] and Păstrăv [124]. Each of these will be presented in this section.

2.3.2.1 BOID, and B-DOING

The Beliefs-Obligations-Intentions-Desires (BOID) architecture [15] extended the BDI architecture model [128] by explicitly adding obligations [15]. Obligations are must norms. A person being obliged to do something means that they must do something. Furthermore, agents are able to violate norms in this architecture, meaning that if a conflict arises between a norm (obligation), and an action (intention) for example, the agents are able to resolve that conflict by either choosing a different action or violating the norm.

But it is exactly in there where one major drawback of BOID lies. The agents resolve a conflict in a fixed manner based on their pre-defined type. This is that regardless of the the situation the agent will always make the same decision. For example: One type is called the *selfish-agent* which will always put desires over any norm (obligation) and thus violate it, whereas the *social agent* type will always put norms (obligations) over desires regardless of the situation. Consequently, the situational nature of real life [12] is not included or at least very hard to include within BOID [15].

The agents cannot take the norm importance into account. Norm importance means that certain norms can be more or less important to the agent. This makes it more or less likely that the agent violates a norm. For example, if a norm does not align with the agent's values or is against the current drives of the agent, the agent is more likely to violate the norm. A lot of decisions that we (humans) make in real life are dependent upon the context in which we make them, which includes our reasons for breaking or following a norm [81]. This is a crucial element for supporting policy-makers, as people will make different decisions in different situations. As described in Chapter 11 in [12], agents might have various different reasons for breaking norms, not all of which are depend on their other desires or intentions, but sometimes simply how much they like the norm, or who instituted it [81]. Furthermore, no other forms of norms are included in BOID [15].

The B-DOING architecture, introduced by Dignum et al. [39], outlines a framework where norms are explicitly differentiated from obligations. Norms and obligations both arise from agent interactions, yet differ in their nature: obligations result directly from agent interactions, often formalized in agreements, contracts, or defined interaction protocols. Consequently, obligations are agent-specific and include penalties for non-compliance.

Dignum et al. [39] categorize norms as social norms, functioning as behavioral standards within groups, organizations, or societies. These norms are seen as stable, abstract, and group-inherent, unlike obligations, which are agent-committed through choice, such as an agreement to pay after ordering.

Decision-making in B-DOING [39] involves norms, obligations, and desires contributing as potential goals. Each modality (norms, obligations, desires) is ranked separately—social benefit for norms, punishment cost for obligations—and then integrated into the agent's utility function, weighing personal utility, social welfare, and cohesion. Goals are selected through heuristics, such as majority voting or a fixed preference order (obligations over norms, norms over desires), providing structure even amid conflicts.

However, several issues exist. One of the main issues is heuristics which are used to determine the behavior of the agents)mentioned by the authors are very vague. This makes it unclear how potential conflicts between the different drives are resolved, and which ones are prioritized. Furthermore, since the architectures is using a utility function, it suffers from the drawbacks that utility functions have, as presented in the previous sections.

2.3.2.2 Values & Motives

The work by Heidari [65] and Heidari et al. [66] is representing social norms and values (the Schwartz value system [138]) explicitly. Furthermore, the concepts of watertanks, which originates from Dörner et al. [40], is introduced. The idea is that for a given motivation, e.g. values in this case, the satisfaction of that motivation constantly decreases and actions have to be taken by the agent to act upon that motivation again. This means that the agent can deliberate about the effect of norms on their values, and which actions to take next in order to adhere to certain values. Furthermore, Heidari [65] takes norm internalization into account, meaning that if norms are internalized by an agent, they are followed automatically,

The purpose of the work of Păstrăv [124] was to develop an agent deliberation architecture for tackling policy-maker support in socio-ecological systems. They explicitly model social norms and policies (legal) norms. Besides these, the classical notions of goals (states the agents wants to achieve) was used. Otherwise, if social norms or policies were active the goals associated with those norms become the preferred goal. The agents then act upon it, if it is possible to achieve that goal. Furthermore, Păstrăv [124] assumes that the social norms are already internalized in their proposed architecture, while the policies are not. This is a difference to the work from Heidari [65] who also looked at norm internalization. This difference in internalizations is crucial, as Păstrăv [124] assumes in their work that when there is a conflict between social norms and policies, the agents will chose to adhere to the social norm and violate the policy, due the internalization of the social norm. Values are used in their architecture to decide between multiple motivations, proving a preference ordering.

2.4 Verification and Validation of Agent Deliberation Architectures

To support policy-makers in their decision-making with social simulation models, it is important that they *trust* the model. This "puts high requirements on the credibility and validity of the generated results" [91, p.331]. With verification and validation of the simulation model, the trustworthiness of the simulation is increased, as well as ensuring the quality of the generated results [91].

Verification and validation (V & V) is currently a hotly discussed topic in the social simulation community. Verification of simpler models, such as evaluating who gets child benefits, is straight forward. If the criteria are met, the benefit is given, and thus it is fairly easy to see if the model makes the right decision. Validation however, is already in simpler systems not easy, as it has to be ensured that the model makes the correct decision in all possible situations.

This becomes even more challenging in more complex and high level systems. For example, if a model is used to assess the fair distribution of goods. Here the term fair is very abstract, and very context dependent. What is fair in one situation might not be fair in another situation. Thus, it is not always clear if the model shows a fair distribution of goods, and thus behaves correctly as intended.

Also, using empirical data for validation is not straightforward as well. The main issue here is what to do when the results of the model are not completely (100%) matching the empirical data, but rather are in some neighborhood or margin of them. For example, in the ASSOCC project [37], the infection curve was not exactly matching reality in terms of exact numbers but followed the trend when infections were rising, when the curve turned and when infections started to slow down or die out. The question is now if this is good enough for the purpose of the model [38].

Furthermore, it is also possible that no real-life event or data exists for the social simulation model to be compared against. As such, verification and validation for social simulations is different compared to the social sciences and the natural sciences, which have data or real-life events for comparison.

It is clear now that verification and validation are highly important for social simulations for policymaking but yet highly challenging when dealing with complex social simulation models required for policymaking. This is why in this section, we will present and discuss existing work on verification and validation of social simulation models.

2.4.1 Verification

According to Gilbert & Troitzsch [55, 91], verification means that the model is working according to the expectation of the modeler [55]. In this sense it evaluates, if the model was build 'right' [6, 91]. This means that one thing that the verification is doing, is to look at the code of the model and to see if it is correct [6, 91], such as being bug free. Furthermore, verification involves controlling that the model is working and matching the underlying theoretical specifications [6, 91]. This means that verifying a model involves evaluating the correspondence of the model with its conceptual model [6, 91].

This goes in line with the arguments presented by David et al. [27]. The authors argue that verification means assessing the adequacy of the conceptual model with the computational model [27, 50]. Adequacy here is a vague term. To further specify this term, the authors say that in situations where the outcome is not known beforehand, such as in the case of investigating potential effects of policies, it is necessary to go beyond making sure that the executable model is doing what it is planned to do [27]. According to the authors it is further necessary to see that for a specific range of inputs, the results of the models are according to the modeler's (or stakeholder's) expectations [27]. In many ways this can be related to plausibility and face validity proposed by van den Hurk et al. [74], where it is checked if given the input range of the parameters, the results of the model make sense (at face value), and are according to expectations. Similarly to the previous paragraph, the authors also argue on a code level that good coding practices should be adopted [27].

2.4.2 Validation

Validation is a crucial part, as it looks at, according to Gilbert & Troitzsch [55, 91], how well is the model matching reality [55, 91]. In other words, validation looks at if the 'right' model was build [6, 91]. Consequently, validation looks at the behavior of the model and how well the model fits its intended purpose [91]. In this sense validating a model is "the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model" [118, p.719].

Here, for social simulations, accuracy has to be understood as a relative term, because models are only *abstractions* of reality, meaning that their cannot be a 1:1 mapping of reality in the model. To determine what is accurate enough, Sargent [132] suggests that a model is accurate enough, and thus valid, if it is in an acceptable range with regards to the purpose of the model, and reflects the targeted real world behavior in a satisfying way [91, 132].

Again this does not make it more concrete, as it is very subjective, and strongly depending on the modeler's judgment to see if it is acceptable or not. One way to tackle this challenge, with regards to agent deliberation architectures, could be to implement the architecture in multiple, e.g. hundreds of models, and then look at the agent's behavior and see if any trends in the normative behavior can be observed which then can indicate if the agent deliberation architecture is suggesting plausible behavior. However, this might not always be feasible as it requires a lot of work, and identifying key measurements to analyze the normative behavior of the agents.

This does however also not solve the problem of the need of existing real world data. Traditionally, models are validated by comparing the results of the models to the behavior of the target system, i.e. real world data [91]. But not every potential use case modeled does have real world data available. A way to solve this, is to perform a comparative validation [91].

Comparative validation means to compare a model to another existing model with the same purpose [91]. An example of this, is the validation process of the ASSOCC model [37, 91]. The ASSOCC model was compared with the model from Ferretti et al. [46]. The validation was considered to be successful if the models were to show similar behavior. Now again the term similar behavior can be vague. Usually it is referred to showing the same trend in the results but the exact numbers do not have to match. This was the case for the comparative validation of the ASSOCC model [91].

While comparative validation can be very useful, it has some challenges. The first challenge is to align the models and make them comparable [91]. This includes the parametrizations as well as the assumptions made in the model [91]. Depending on how different the models are, this can be very hard. The next challenge are the input variables and parameters of the models. Different models can have different inputs and parameters. To compare different models, it needs to be clear which variables from one model match to the variables in the other model [91]. Finally, the different scales of the models, and their general modeling paradigms need to be taken into account when comparing two models [91]. For example, the ASSOCC model [37] is a behavior-based model, while the model of Ferretti et al. [46] followed a equation-based mathematical paradigm [91]. Furthermore, the scales of the model are also to be taken into account. For example, the ASSOCC model only consistent of a couple of thousand agents [37], while the other model was able to simulate millions of agents [68].

A promising approach to deal with the absence or strong limitation of real world data has been developed by Kleijnen [86]. According to Kleijnen, systematic experimentation should be performed to gather data from the model that is suitable to asses the quality of the model [86, 91]. One method that can be used for this systematic experimentation is the *one factor at a time approach* enabling a what-if analysis of the model [86, 91]. Such a factor to be changed to assess the quality of the model can be for example the addition of a new norm to see how the agents in the mode deal with that or the increase of the action space of the agents (i.e. more actions are available to the agents), but importantly not both at the same time.

A novel validation approach was recently developed by van den Hurk et al. [74]. They propose a "structural validation approach for assessing an explanatory social ABM by integrating face, trace, and event validity" [74, p.4]. This is a qualitative validation approach and not a quantitative one [74]. Face validity means that the behavior of the model and the underlying processes are plausible for their purpose [74, 87, 131], based on the common sense of the researcher [74]. This can colloquially be rephrased as the behavior of the model and the underlying (theoretical) processes should make sense when looking at them at face value. In many ways this goes back to the verification by Gilbert & Troitzsch [55] arguing that the model should behave according to the modelers expectations, as the same common sense and comparison to expectations process is involved in this step when looking at face validity.

The next step, trace validity, is a crucial part of the validation process, as here the (macro-level) outcomes of the model need to be explained by the (micro-level) behavior of the agents. In many ways this involves data storytelling and explaining the results with the behavior of the agents. It is important here that the results of the model must *only* be explainable by the behavior and micro level interactions of the agents, and not by the implementation choices, or (even worse) bugs in the model [74]. The ASSOCC project did this for example for the track and tracing app scenario [76, 91] by explaining the results rigorously with the behavior of the agents and telling a story with the resulting graphs. Hurk et al. [74] argue that "Trace validity, especially with face validity, is also essential when new, unexpected outcomes emerge". [74, p.7]. This is important when considering the fact that the outcomes of the indented policies are not known or go against the expectations of the policymakers.

Finally, event validity is the most difficult one to achieve, as it involves relating the model to real-world events. This is very difficult to do beforehand if similar situations have not occurred yet. To solve this challenge, Hurk et al. [74] argue that "The events modeled do not need to have occurred but should be relatable to realistic scenarios [...] When combined with face validity, this approach allows reasoning about realistic scenarios, even if they have not actually happened" [74, p.8]. Finally, the authors argue that qualitative validation increases the explainability of the model [74] which is crucial for policymaker support.

2.5 User Interaction Tools

In order to support policymakers in their decision-making with social simulations, it is crucial for them to understand *why* something is happening in the simulation, e.g. why some agents violate a norm (policy), such as a curfew, or why the behavior of the agents changes after the introduction of a new norm (policy), and not just that it does. Being able to do this also increases trust in the simulation. Consequently, trust does not solely come from the verification and validation process, but rather from a combination of verification and validation, and the possibility to explore and modify the simulation by the policy makers (users) of the simulation. This goes beyond the interfaces provided by the social simulation tools, such as Netlogo [162] and Repast [117], because they focus on debugging and quick use of the expert user (modeler) rather than the use of non-expert users (policy makers).

When looking into the literature, one has to be very careful however, as often different stages of the modeling process have been targeted. Much effort in the social simulation community has been put in participatory modeling approaches, see e.g. [1, 100]. Participatory modeling focuses on developing the models together with the stakeholders. While this is also very beneficial in creating a shared understanding of the model and support the policy maker (as one potential stakeholder), such approaches mostly only focus on the first step of the modeling process shown in Figure 2.1, namely building an abstraction from reality.

However, When focusing on steps two and three in the modeling process (see Figure 2.1, the support of tools in the social simulation community becomes very thin. Not much work has been done here. This is for example pointed out by Ahrweiler et al.[1], who also used a participatory modeling approach report in their feedback from stakeholders that "[...] new visualization and interactive technologies can help to make simulation results more accessible to stakeholders" [1, p.8]. This shows that not much work has been done here. Nonetheless, there are a few works to be pointed out.

A notable earlier work is the work by Broekens et al. [14]. They focused on self-explaining BDI agents in their work and which kind of explanations are most natural for users. This form of explanation goes in line with the work done by Harbers et al. [64] who also worked on self-explaining BDI agents. By traversing the goal-believe-action tree bottom up their explanations from the agent's point of view can be summarized as: I (agent) $< action >$, because I $< goal >$ and I $< believe >$. This is very interesting to us, as we can use this approach later to explain the behavior of individual agents, which provides a great source of insight for policymakers, in terms of how individuals might react to potential policies.

Another important work is the ASSOCC project [37]. In terms of accessibility for the user, this project provided a huge step forward. This project became very complex in terms of agent complexity, and parameters that can be adjusted. As a consequence of that, the NetLogo [162] interface became very complex, see Figure 5.4[1].

Figure 5.4 is a composite image which means that the complete interface did not fit on one computer screen. This makes it not only very difficult for expert users (modelers), but completely impossible for policymakers (non-expert users) to gain any meaningful understanding from it. This was solved by utilizing the power of games engines. Specifically, the Unity[2] game engine was used to make an interface of the simulation [123]. This interface can be seen in Figure 2.5.

The figure shows that this interface is much simpler and easier to understand while retaining the core functionality and information of the original NetLogo [162] interface (see Figure 2.4). The buttons at the top, and at the bottom highlight that there is more information to be shown but currently hidden. This is a vast improvement compared

[1]The figure is to show the complexity of the interface, and not inteded to be fully readable. The non-readability is the point that is made here.

[2]https://unity.com/

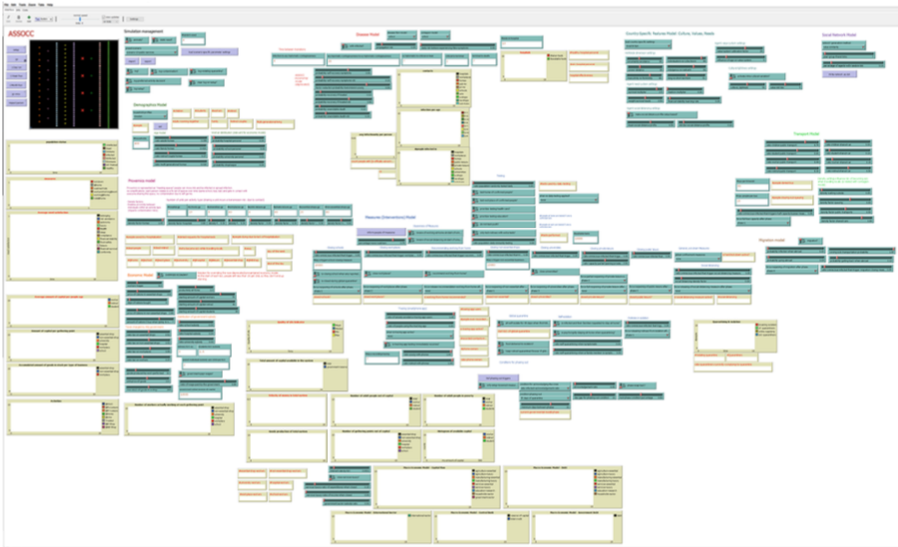


Figure 2.4: NetLogo interface as a composite image of four individual images, taken from the ASSOCC book [123]

Figure 2.5: Unity interface, taken from the ASSOCC book [123]

to the NetLogo [162] interface, as such functionality is not available there. Having a cleaner interface also makes the communication with the policymakers easier, as they can access the simulation quickly and understand what is going on by being provided with essential information, and protected from unnecessary complexity and potential information overload. For example, the flight lines in Figure 2.5 represent the agents moving from location to location, and the red color indicates that they are infected (gray otherwise). This provides easy access for understanding how COVID-19 spreads between different locations. Furthermore, the buildings are colored red, if at least one member of that building is affected. This provides further understanding of the current state of the spread of the virus.

Further use of the Unity game engine has been shown by Păstrăv [124]. Their work was done in Unity with the later involvement of the user in mind. While they provided a very modular implementation, the interface was only text-based. It does not go beyond the use of the Unity editor. While this text-based approach is very limiting in terms of user support and interaction, it still shows the potential and ability for using the Unity game engine in supporting policy makers.

The work of Heidari [65] should be mentioned as well, because they ensured a very rich model where the modeler can see many more aspects of the agents compared to Pastrav [124]. But unfortunately the work of Heidari [65] did not focus on user interaction, and thus no support for the model was developed to find the rich information that is within the model. To this extend, it would be wishful to create a combination between both the work of Heidari [65] and Păstrăv [124]

Finally, the GAMA platform [147, 148] is also an important work to mention. The GAMA platform provides an open-source environment for developing spacial agent-based models. It provides modifiability of the agents [148], as well as various visualizations that the user can chose from the for a simulation run [59, 148], such as different kinds of maps. There are several problems however. The first one is that the modification support for non-expert users is not well developed. Users can modify agents, but the way they have to do it, is still very technical, no interface support is provided. The second issue is that there is no special focus on norms. The focus is only on the agents, and not on the norms. This means that the norms are not easily modifiable but rather only exist in code. Furthermore, there is no way to model norms to later add them in the simulation.

2.6 Conclusion

This chapter has introduced topics relevant for the research in this thesis and identified gaps and limitations.

The discussion began by situating legal norms within a broader web of social rules, as these norms do not function in isolation but interact with other social dynamics. Modeling legal norms and, to some extent, habits, establishes a groundwork for integrating multiple normative perspectives, as they need to be understood first. Furthermore, existing works in deontic logic (e.g. [109, 159, 156] and the ADICO grammar [25] were introduced to provide an understanding for formalizing norms in agent-based social simulations.

Two primary approaches in norm modeling—implicit and explicit—were identified. The ASSOCC project [37], as a powerful example, showed how social theories, such as values, motives, and affordances enhance agent decision-making, but was still integrating the norms into the actions of the agents. Recent works by Heidari [65] and Păstrăv [124] offer models in which norms are explicitly represented, an essential step towards allowing policymakers to modify norms independently. However, existing models have yet to incorporate diverse perspectives on norms, a gap this thesis aims to address.

Since the verification and validation (V&V) in social simulations is challenging, various approaches were reviewed, such as comparison to real-world data or comparison with other models [91]. Promising approaches from Kleijnen [86] and van den Hurk et al.[74] were considered, which, in combination, support Epstein's [42] view that modeling should reveal core dynamics—for this thesis, the dynamics of normative decision-making.

Although limited support and existing work were found in the literature for a potential interaction tool, the ASSOCC project [37, 123] and Păstrăv's work [124] provided the most relevant basis. Both use the Unity game engine, with ASSOCC [37, 123] demonstrating the potential support it offers for non-experts users, and Păstrăv's work [124] achieving the necessary modularity. However, the norm specification is only text-based so far with no additional support for the non-expert user. Addressing the limited non-expert user support and existing work on a potential interaction tool is the other gap in research that this thesis aims to address.

Chapter 3

An agent deliberation architecture for dealing with different interpretations of norms

3.1 Introduction

Now that we have a better understanding of the field of agent-based social simulations, and existing research in normative behavior, we can look at the purpose and objective of our research to derive the necessary requirements for our agent deliberation architecture[1]. Recall, our objective is to support policymakers in their decision-making when designing and evaluating the potential effects of intended policies. This requires human-like realistic behavior from agents in a particular context. Human-like realistic behavior can mean a lot of things, here we focus on the role that norms play in the human-decision making process. We focus on norms, because they describe what is considered appropriate behavior in a certain context and aim at assuring the interests and values of groups or the society as a whole [102]. Norms are a crucial part of human behavior, and influence it in a variety of ways and multiple levels [34, 78, 81, 82]. The concept of norms encompasses different types of acceptable behavior. Social norms are informal context dependent (e.g. culture or society) expectations of peoples behavior [102], e.g. formal greeting with a handshake in Germany[2]. Legal norms (i.e. laws, and policies) are institutionalized norms [102] imposed by an authoritative body, e.g. a government or other policy maker [82]. We want to note here that there

[1]The research presented in this chapter is based on and partially published in our following papers: [78, 81, 82]

[2]Here Germany provides the context. Note that in other countries the correct behavior would be for example bowing down.

are many more kinds of norms than social or legal norms (e.g. moral norms [102]). Hence, we refer to the existing body of research on norms, see e.g. [9, 12, 22].

We focus on legal norms in this thesis. Legal norms are used to achieve desired situations [102], and provide the way for policymakers to influence the behavior of people[1]. For example: During the COVID-19 pandemic the desired situation for the policymakers was to reduce the number of infections (i.e. flattening the curve). To achieve this, they wanted to implemented policies, such as curfews, to reduce the spread of the virus and achieve that desired state of reduced infections.

Two main challenges for policymakers arise from this which we are solving. The first one is that when dealing with norms, each person or groups of people engages differently with a (legal) norm to make sense of what it means for them, and how it impacts their behavior or their life in general. This cognitive process of sense-making is what we call *different perspectives on norms*. This means that people only take the parts of the norm into account which are relevant for them.

This leads to the second challenge, namely that policymakers need to be able to assess their (the norm's) effectiveness [81], which requires them (policymakers) to able to gain meaningful insights from using social simulations. Based on these insights policymakers must be able to modify the norm, as it might be possible that the norm did not have the intended effects to reach the desired state, e.g. the curfew did not flatten curve and therefore the policymakers might want to change the time of the curfew or try additional measures.

Figure 3.1 illustrates the challenge how a norm (here a size-based restriction on the number of guests in a restaurant) is influencing people differently, and everyone wants to figure out what it means for them. They do this by **reasoning** about it based on their own **subjective** characteristics (which is represented by a different color that each person has indicating that they are all different).

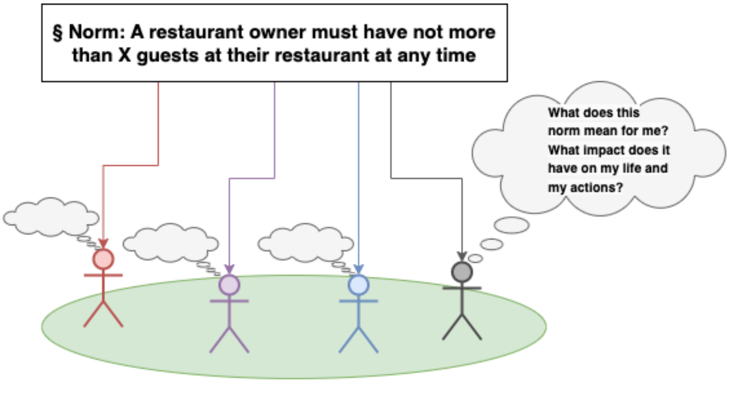


Figure 3.1: People subjectively reasoning about the norm, trying to find what it means for them, with colors indicating that everyone is different and thus differently impacted by the norm

[1]Compared to social norms, which arise more from the interaction of people rather than being imposed from the outside

The perceived subjective impact of the norm on the individual, represented by the different colors and arrows, is exactly what we are required to model for the agents to be able to reason about the potential impact that a norm has on them. Furthermore, we require this connection to be concrete for policies to have a direct influence on people's behavior to be suitable for policy support but also abstract in the sense that our agent deliberation architecture should be suitable for general policy support, and hence specifiable for different scenarios rather than specifically tied to one scenario or purpose, such as the agent deliberation architecture [75] from our ASSOCC project [37] which was specifically developed for the COVID-19 pandemic. The requirements for developing this agent deliberation architecture will be detailed in the next section.

The remainder of chapter is organized as follows: We start with outlining the requirements for our proposed agent deliberation architecture (Section 3.2). In Sections 3.3 and 3.4 the concepts related to different perspective on norms, and norms are formalized. This will be brought together in Section 3.5 where we present our proposed **perspective-based agent deliberation architecture (PBADA)**. Finally, in Section 3.6, we conclude our work presented in this chapter, and outline potential future work, since this architecture is the first step of a potential new generation of norm-realistic agent deliberation architectures for policy support.

3.2 Requirements

Based on our objective to *integrate different perspectives on norms into the agent's deliberation process and make them modifiable*, we derive the following three groups of requirements:

1. Norm realistic behavior, including different reactions to norms and subjective reasoning about norms.
2. Modifiability of the norms
3. Scalability

The norm realistic behavior requirement entails the potential reactions to norms that we see humans display: obey the norm, obey and circumvent the norm (motivational component of the norm), and violating the norm[1] [20, 36, 78, 81, 82, 120]. To decide which action to take, the agent needs to be able to **deliberately** decide what they perceive to be the best decision. Deliberation is necessary here, as otherwise assumptions have to be made about the agents, and their behavior in all potential situations has to be defined beforehand (i.e. having a rule-based architecture). This is not only very dangerous, as the assumptions can be completely wrong and thus resulting in misleading simulations, but it is also very rigid and does not allow for any (quick) modifications (on the fly).

[1]Note that this entails all possible reactions, e.g. people can obey the speed limit, violate the speed limit, or obey the speed limit but start driving earlier to make up for the lost time due to the speed limit (motivational component)

To enable the agents to deliberately decide what they perceive to be the best decision, they must reason **subjectively** through their perceived consequences of the norms. With subjectivity we mean that they only engage with the parts of the norm which are relevant for them.

We thus operationalise norm realism (subjective norm reasoning) through explicitly needing to include deliberate and subjective reasoning. The agents need internal drives of behavior to decide which action to pursue, which in our case are needs. Consequently, the agents also need multiple behavior possibilities, i.e. actions, to act upon their needs to satisfy them. Furthermore, the agents need a subjective (incomplete and perspective sensitive) representation of the world to understand which actions are affordable (currently available to them). Together these elements, in addition with the memory of the agent which is required for the scalability of the simulation, are making up the internal world of the agent. Finally, agents need to be able to interact with the environment and other agents (at the least react to the behavior of other agents).

The **modifiability of the simulation requirement** enables the policymakers to really use the simulation as a sandbox for their decision-making process. For them to try out different norms, they must not only be able to add new norms to the simulation, e.g. a curfew to see if it flattens the curve, but also to modify existing norms, e.g. the time of the curve after the norm has been added. Since the policymaking world is a fast paced dynamic one, policymakers must be able to add and modify norms by themselves on the fly [82].

For policymakers to be able to track the potential effects of norm changes, assess their potential impact, and establish a clear causal chain, there must also be limitations to the modifiability of the simulation. This means that the agents (e.g. their needs and actions), and the world itself (e.g. the objects in the world) are not modifiable. Keeping them fixed, also ensures a working simulation. Meaning that the simulation does not suddenly break, because the policymaker did add an incomplete action for example.

The **scalability requirement** supports policymakers gaining meaningful insights in a reasonable amount of time. The simulation must be able to handle a significant amount of agents[1] for policymakers to be able to gain meaningful insights. If the simulation can only simulate a few agents, no implications on the potential effects of norms can be drawn for a country for example.

The simulation must run in a reasonable amount of time (e.g. minutes or at maximum hours), and cannot take too long to run. Otherwise, the simulation is of no practicable use for the policymakers, as they need quick insights, and cannot wait days or weeks for a simulation run to complete to then make a change and wait for the same amount of time again.

3.3 A perspective on norms

Based on the requirements outlined in the previous section, we are now formalizing what we mean by different perspectives on norms, and thus address the requirement

[1]What significant means depends on the purpose of the simulation

regarding the subjective engagement with norms. To enable agents to engage differently with norms, we need characteristics that actually make them different. Furthermore, since normative environments are dynamic, meaning that existing norms can change or be removed, or new ones can be added, the agents need to be adaptive in their behavior. To account for this, we use the notion of *needs* as the underlying motivators for the agents to make their decision. Need satisfaction is dynamic, and thus the agents can engage with a norm based on their current need satisfaction. This is also why utility functions [30] are not suitable, as they have a fixed reward in terms of the utility they provide, while the needs are providing a variable reward in form of need satisfaction, meaning that if a need is satisfied, an action targeting that need is not that attractive. Furthermore, we need to connect these internal personal drives to a more abstract system to guide the agent in resolving conflicts, for example if two actions are equally attractive. We do this by the priorities in values that an agent has. Finally, we need a way to connect the environment to the potential actions of the agent. We do this by using the concept of affordances [36]. Furthermore, we introduce the notion of social affordances, as people go differently through the world and decide their action based on their subjective social meaning.

We successfully used the concepts of needs, values, and affordances in our ASSOCC project [37] with a detailed description by Dignum in chapter two [36] of our ASSOCC book [36]. Furthermore, parts of the described elements, such as motivators in Di Tosto & Dignum [32] or values in Boshuijzen - van Burken et al. [11] , have previously been used in [66, 71, 122, 124, 150, 151] further adding to the validity and necessity of using these elements.

Bringing this all together and assuming a total set of $\mathcal{P} = \{p_1, \ldots, p_n\}$ perspectives in a simulation, we define a perspective $p_i \in \mathcal{P}$ as follows:

Definition 1 (Perspective)
A perspective is specified by their needs ($\mathcal{NE}$), available actions ($\mathcal{ACT}$), social affordances ($\mathcal{SOCAFFS}$), and priorities in values ($\mathcal{PRIOV}$).

We are going through each part of this definition starting with the needs.

3.3.1 Needs

For the agents to deliberately reason about norms, we need to provide them with motivators for their behavior on which they can base their decision upon. Since the agents are operating in a dynamically changing environment, where norms can be modified or added, these motivators cannot be hard rules, but rather have to provide guidance and inform behavior in such a way that at certain points in time some behavior is more important, i.e. urgent, to do, compared to other behavior. In other words, they determine which behavior is *salient* at specific points in time.

Furthermore, they allow for some kind of equilibrium oriented, i.e. *stable*, behavior overtime. While Dignum [36] argued that this is important in crisis situations, we go one step further and say that this is important in general when trying to model human behavior for policy support. It is crucial to not only look at the short-term effects of policies (legal norms), but also on their potential long-term effects. It might be possible that some policy interventions are beneficial in the short-term but potentially harmful in the long-run.

To meet these requirements, we use the motivation theory from McClelland [96] as our basis. The following four motives are distinguished by McClelland [36, 96] (recall Section 2.2.3 in Chapter 2).

- Achievement
- Affiliation
- Power
- Avoidance

Within this theory, McClelland [96] suggests that all humans have all four motives at all times but with varying degrees of importance. This means that we prioritize certain motives over others. For example: a restaurant owner can be more achievement driven, and is thus seeking states in which they feel that the progress, e.g. having more money. Another example could be: a guest that has a high priority in the affiliation motive might go out a lot to socialize with their friends.

When using the motives by McClelland [96], we have one more challenge to solve, the abstractness of these motives. We have already seen (in Section 2.2.3 in Chapter 2) for the achievement motive, for example, that it is very personal what achievement means. Consequently, it is not a trivial task to directly connect these motives to concrete actions. This is why we concretize them into ***needs***. For example for the restaurant owner, the achievement motive could be concretized to a financial need, as they want to make money with their restaurant. These needs can then be connected to concrete actions to satisfy them. An example depiction of this can be found in Figure 3.2.

To operationalise the needs, we use the watertank model introduced by Dörner et al. [40] which was successfully implemented in social simulations by Heidari et al. [66], and us in [37, 88]. In these models, each need is represented as one watertank.

It is important to note here that we adopt the formalization from the work of Heidari et al. [66], and our work in [37, 88]. The first step is to define the range of the watertank which is $[0;1] \in \mathbb{R}$ with 0 meaning the watertank is empty and 1 the watertank is completely full. The current fill level of the watertank is consequently a value within that interval, and denoted as $\ell \in [0;1]$. To enable the agents to use these watertanks as motivators for their actions, and provide them with some indication on what is more salient for them to do right now based on their past activities and current situation, we need three more elements, namely the threshold, the depletion rate, and the urgency. We define the threshold as $\mathcal{t} \in [0;1]$ and it describes the minimum fill level of the watertank that is required for the agent to consider that need satisfied. Furthermore, the higher the threshold, the more important the need is for the agent. The depletion rate $\mathcal{d}$ is also defined on the real interval $[0;1]$ as $\mathcal{d} \in [0;1]$. It describes the satisfaction loss of the need, i.e. the depletion of the watertank, at each timestep if the need is not attended to by the agent. Formally speaking, at every timestep t in the simulation we are applying the following to each watertank i:

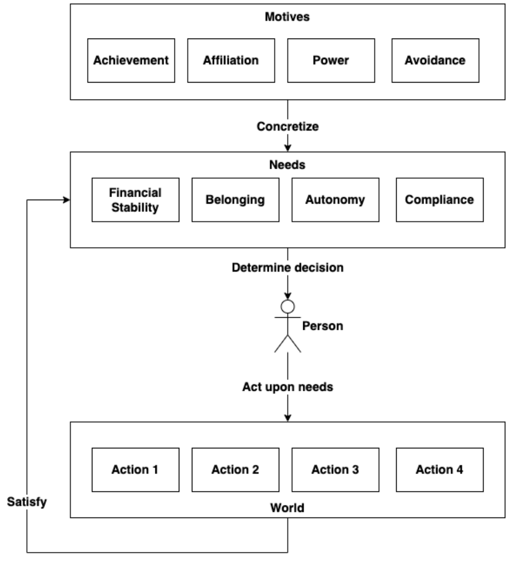


Figure 3.2: Example depiction of how we connect motives to behavior. The motives and needs above the person are inside their head, an the actions (below the agent) are outside in the (physical) world.

$$\ell_i^t = \ell_i^{t-1} - d, \tag{3.1}$$

with i denoting the watertank, and t the current timestep.

The final element that we mentioned that we need now is the urgency of the need. The urgency u describes the salience of the need, meaning how important it is for the agent to satisfy that need. The higher the urgency, the more important it is for the agent to take action to satisfy that need. Formally, the urgency $u \in [0;1]$ describes the difference between the current fill level ℓ and the threshold t. If the current satisfaction level is higher than the threshold, then the need is satisfied and the urgency 0. We calculate the urgency of the watertank i at every timestep t as follows:

$$u_i^t = \begin{cases} t_i - \ell_i^t & \text{if } t_i > \ell_i^t, \\ 0 & \text{if } t_i \leq \ell_i^t \end{cases} \tag{3.2}$$

with i denoting the watertank, and t the current timestep.

A visual depiction of a watertank and the elements can be found in Figure 3.3:

3.3.2 Values

Values are evaluation criteria for events and behavior [129, 139]. Based on the various definitions of values, we can extract a set of three common features [88, 138]. First, values are relatively stable beliefs about desirable end-states of reality that motivate

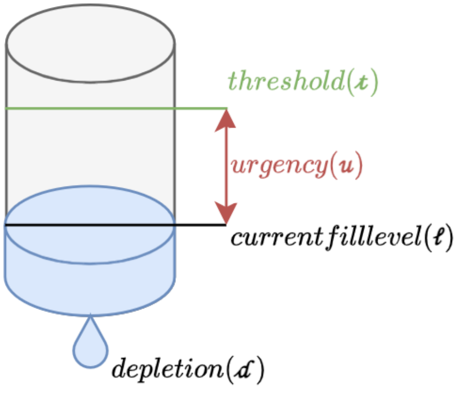


Figure 3.3: Example watertank depiction

action. Second, values serve as general guiding principles for behavior and decisions over a wide range of contexts [3]. Third, a person's values are ordered based on their relative importance to that person [138].

These characteristics show that values can be used to guide behavior in conflicting situations, e.g. two actions are equally attractive for the agent. Furthermore, the last characteristic allows us now to formulate the priorities in values for an agent, such that they have a way to decide between two actions. Additionally, we assume that the priorities in values for each agent stays constant over the time of the simulation run.

We define the priority in values ($\mathcal{PRIOV}$) for each agent as follows:

Definition 2 (Priorities in values (PRIOV))
Given a total set of $\mathcal{V} = \{V_1, ..., V_j\}$ in the simulation, the $\mathcal{PRIOV} = [V_1, ..., V_m]$ is an ordered list of $\mathcal{V}$ with $V_1 \succ \ldots \succ V_m \; \forall V_i \in \mathcal{PRIOV}$, whereby $|\mathcal{V}| = |\mathcal{PRIOV}|$ holds.

We use the priority in values ($\mathcal{PRIOV}$) in the current version of our presented agent deliberation architecture to resolve conflicts between actions. This means that when two actions provide the same perceived need satisfaction gain, the agent uses their $\mathcal{PRIOV}$ to decide which action to take, i.e. the one that aligns the most with their values. When two actions, for example going to the park and going to the restaurant, are equally as attractive from a needs point of view, the agent looks at the values, and decides for the one that is aligning most with them. Aligning most in our case means that the agent prefers the action that promotes the value with the highest position in their $\mathcal{PRIOV}$.

The reason for doing it like this, is that the $\mathcal{PRIOV}$ are a qualitative preference ordering and not a quantitative one. It is not clear to what extend V_1 is more important than V_2 by the agent. Since we have no numerical operators, we cannot say that action promoting values V_2 and V_3 is preferred over an action promoting just V_1. This holds for any combination, because we have no numbers to add or multiply and compare against each other. Therefore, we look at the promoted values by the action, and chose the one that promotes the value V_i that is the highest in the $\mathcal{PRIOV}$ of the agent.

In case of a norm action conflict, i.e. the adhering to the norm is as equally attractive as violating the norm, we argue here that in general people want to follow norms when possible, and thus if the action is not clearly better than adhering to the norm, people will follow the norm. For example if adhering to a speed limit is providing the same perceived need satisfaction gain as violating the speed limit, the agent will adhere to the speed limit, because violating the speed is not clearly better than adhering to the speed limit.

When two norms are in conflict with each other and only one can be adhered to and the perceived need satisfaction gain is the same for violating either of them (given the respective actions), the action that aligns more with agents values is chosen.

The value system we use for this purpose is the Schwartz Value system [138], see Figure 2.3 in Chapter 2, which has seen extensive amount of attention and empirical validation [130]. Furthermore, it has been used in various social simulations, see e.g. [37, 65, 124].

A big advantage of using the Schwartz value system [138] is that the values are ordered on the circumplex in such a way that values which are related are next to each other and values which are opposing each other are further away from each other. This allows our agents to avoid potential conflicts when looking at an action. One action cannot promote a very important value of the agent and simultaneously demote another important value of the agent. Values placed close to one another in the circumplex are considered mutualistic (tend to harmonize with one another) and values placed further away from one another become increasingly antagonistic (are generally in conflict with each other) [88, 138]. The Schwartz values are [138] as follows (starting with achievement and then going counter clockwise on the circumplex in Figure 2.3 in Chapter 2): achievement, power, security, conformity, tradition, benevolence, universalism, self-direction, stimulation, hedonism.

For example: When the agent has a high priority in universalism, the Schwartz value system suggests that the agent also finds benevolence very important, whereas power or achievement are not highly prioritized values. This line of argumentation also holds for the actions. It is not possible for an action to simultaneously promote universalism and demote benevolence, given their relation within the Schwartz value system. Heidari et al. [66] utilized these characteristics in they simulation with values.

Another advantage of using the Schwartz value system [138] is that it has received an extensive amount of attention and empirical validation [130]. This means that we can ensure the realism of the value system used in our architecture to support policy makers and not an arbitrary system which might be far from reality.

3.3.3 Affordances & Actions

The next part of Definition 1 that we are focusing on are the affordances and actions. Since these elements are strongly connected to the object that is used by them, we first want to assume that we have a set of n objects $O = \{o_1, o_2, ..., o_n\}$. Furthermore, when it comes to actions, we assume that an agent has the necessary capabilities to perform the physical action, e.g. the agent is able to walk or drive.

Affordances are a determinant of human behavior [36, 54]. In general, affordances describe the "action possibilities provided to the actor by the environment"[1]. Thus, affordances determine what kind of behavior is salient [36].

The classical notion of affordances focuses on the physical action options that are provided by an object, e.g. the table is offering the ability to sit on it, sit-ability, and put an object on it, place-ability. It contains the actions one is invited to do with it, or normally do with it. E.g. one can saw the legs from a wooden table, but that is usually not thought as of an affordance of a table. In general the set of affordances can be subjective and context dependent. We view affordances as a *relation* between an object and the action possibilities that the object provides, i.e. the object affords the actions [82].

With this view, we can define define the physical affordances of an object as follows:

Definition 3
The physical affordances ($\mathcal{PHYSAFFS}$) of an object (o_i) describe the physical action possibilities, $\{\alpha_{physical}, \beta_{physical}, \ldots\}$, that the object ($o_i$) provides,

$$\mathcal{PHYSAFFS}(o_i) = \{\alpha_{physical}, \beta_{physical}, \ldots\},$$

for any $o_i \in O$ where o_i has the necessary properties suitable for the actions.

Based on this, we can more formally define the physical affordances of the object o_i as:

$$\mathcal{PHYSAFFS}(o_i) = \{A \in \mathcal{ACT}(o_i) | object(A) = o_i\} : O \mapsto 2^{\mathcal{ACT}}, \tag{3.3}$$

with $\mathcal{ACT}(o_i)$ describing the physical actions performed with the object o_i. Formally, we define a physical action $A_i(o_i) \in \mathcal{ACT}(o_i)$ as follows:

$$\begin{aligned} A_i(o_i) = \langle object : o_i, pre : 2^S, post : 2^S, promValues : promV_{A_i} \subseteq \mathcal{V}, \\ demValues : demV_{A_i} \subseteq \mathcal{V}, Affected\_Needs : Ne \subseteq \mathcal{NE} \times \mathbb{R} \rangle, \end{aligned} \tag{3.4}$$

with: object = the object of the action, pre = the pre-conditions of the action, post = the post-conditions of the action, promValues = the promoted values of the action, demValues = the demoted values of the action, and Affected_Needs = the affected needs of the action.

People do not use objects in only a pure physical way. They choose their physical actions based on their role and the social effects of their actions (social actions), e. g. people go to a restaurant to sit at a table together to socialize. This means that sitting together at a table in a restaurant offers socializing. Therefore, we have to move beyond the physical notion of affordances and look at the *social interpretation* of the use of the object next to the physical use. We call this *social affordances*. They describe the social use of the physical object.

[1] https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/affordances, accessed: 02/20/2021

This means that people chose the physical actions based on its *purpose*, i.e. the social action. E.g. the purpose to go to the restaurant is to socialize. This goes in line with the argument by Dignum [35] who state in their work that the purpose "determines the social interpretation [..] of certain physical situations" [35, p.2].

When using this social view on affordances, and using the purpose of an action, we can start to realize the great benefit when creating social simulations in which the persons are supposed to react to changes in the environment [36]. These changes can be physical, but also social (such as new (legal) norms) [36].

We look here at our restaurant example to illustrate this. Guests go to the restaurant to socialize (the social action, i.e. the purpose of going to the restaurant). The new restaurant size-based restriction norm now does negatively impact that possibility, and the guests might decide to chose an alternative action with the same purpose, such as going to the park to socialize with their friends or invite them to their home. Furthermore, this is also important when reacting to the behavior of other agents. It might be possible that as a consequence of the norm, some restaurant owners were forced to close their restaurant. As a consequence the guests cannot go there anymore and need to find an alternative.

This means that social affordances enable the agents to create alternative actions on the fly [36]. Without the use of social affordances one would need to specify all possible behaviors for each type of agent in each possible situation [36]. However, the affordances that are described for each element in the environment (the potential object of an action) will determine the boundaries of the alternative behaviors that can be expected in a simulation [36]. In general, having the agents making use of these affordances will create a more modular and flexible way of specifying potential behavior of the agents which can determine new ways of behaving based on the current situation and available affordances [36].

To now define social affordances, based on the physical affordances defined in Definition 3, we keep the physical object and add the social aspects of time and location, and the perspective of the agent to formulate the social affordances that an object provides. The social meaning of the time (the social time, $socT$) of an object refers to the association with it, e.g. the evening is associated with dinner time. Different times have different meanings and can therefore influence the available social affordances. This is similar for the social meaning of the location (the social location, *socL*), e. g. a restaurant is a place to eat. We also need to look at the perspective of the agent, e. g. being a restaurant owner. The definition of social affordances is as follows:

Definition 4

The social affordances ($\mathcal{SOCAFFS}$) of an object (o_i) are the social effects of the physical actions performed (social actions), $\{\alpha_{social}, \beta_{social}, ...\}$, with the object ($o_i$). They are determined by the social time (SocT), the social location (SocL), the perspective of the agent (p_i) with the associated actions ($\mathcal{ACT}(o_i)$), and the object (o_i) itself. The resulting formalization is then:

$$\mathcal{SOCAFFS}(o_i, SocT, SocL, p_i, \mathcal{ACT}(o_i)) = \{\alpha_{social}, \beta_{social}, ...\},$$

for any $o_i \in O$ where o_i has the necessary properties suitable for the actions.

We can see here that we are using elements which we call *SocT* and *SocL*. *SocT* is the social meaning of the physical time, and *SocL* is the social meaning of the physical location. We can define these mappings as follows, given the set of physical times $T = \{t_1, \ldots, t_n\}$, and physical locations $L = \{L_1, \ldots, L_n\}$

$$\begin{aligned} socT &: T \rightarrow SocTimes \\ socL &: L \rightarrow SocLocs \end{aligned} \tag{3.5}$$

With `SocTimes` being the set of all possible social times in the simulation, and `SocLocs` the set of all possible social locations in the simulation.

To integrate the social affordances now to the actions of the agents, and allow for alternative behavior on the fly, we translate them to the purpose of the action. To do so, we adapt Equation (3.4), and add the purpose element to it as follows:

$$\begin{aligned} A_i(o_i) = \langle object : o_i, purpose : \mathcal{SOCAFFS}_{A_i} \in \mathcal{SOCAFFS}, \\ pre : 2^S, post : 2^S, promValues : promV_{A_i} \subseteq \mathcal{V}, \\ demValues : demV_{A_i} \subseteq \mathcal{V}, Affected\_Needs : Ne \subseteq \mathcal{NE} \times \mathbb{R} \rangle \end{aligned} \tag{3.6}$$

with: object = the object of the action, purpose = the purpose of the action (i.e. the social affordances of the action), pre = the pre-conditions of the action, post = the post-conditions of the action, promValues = the promoted values of the action, demValues = the demoted values of the action, and Affected_Needs = the affected needs of the action.

Roughly speaking we achieve the following by doing it this way. Assuming that both, going to the restaurant and going to the park, socially afford to socialize with friends, we can ascribe both of them the purpose to socialize. When for example the action to go to the restaurant is not available anymore, the agents can on the fly find another action to socialize and satisfy their needs (the reason they initially chose to go to the restaurant), such as going to the park.

3.4 Norms

To formulate norms and also allow for them to be adaptable during the simulation, we need a framework that defines clear and modular elements of a norm which can then individually be targeted. Furthermore, this needs to be structured in such a way that it is easily understandable by the user. The narrative policy framework [135] focuses on the role that narratives play in the public policy making progress which can help for user understandability but lacks the necessary precision for a formalization that can be used in agent-based social simulations.

The framework that we use is the widely applied ADICO grammar (also referred to as the institutional grammar) by Crawford & Ostrom [25]. The grammar provides a general format for rules [144], and was specifically developed to analyze institutional statements [25], such as legal norms. The goal is to understand action situations, and "provide a detailed depiction of what actions are allowed, permitted, and forbidden under specified conditions and often with specific sanctions for actors" [142, p.79]. Thus, it offers a way to identify the core elements a policy is comprised of [142]. This specified way is very important for our case, as policies (laws) need to be very precisely formulated. Siddiki et al. go as far to say that the purpose institutional grammar (ADICO grammar) is "[...] analogous to genetic codes in living cells [...]" [142, p.81]. Furthermore, this grammar has been used in a variety of simulations, see e.g. [53, 144], adding to its validity and applicability for agent-based social simulations. The ADICO grammar consists of the following five elements [25, 82].

A Attribute: The person(s) responsible for adhering to the (in our case) norm

D Deontic: The deontic element (obligation, prohibition, permission)

I aIm: The targeted action of the norm, e.g. the action which is forbidden

C Condition: The circumstance under which the action is e.g. forbidden

O Or else: The punishment for violating the norm

To allow for the subjective reasoning about norms, we also need to extend the ADICO gramar. This is important, as we need to make this otherwise hidden subjective reasoning explicit, such that the agents can reason about it. To do so, we firstly expand the ADICO grammar to our ADICDlRO framework, which we introduced in [82]. Our framework adds the deadline condition (Dl) describing the norm takes effect in the simulation (i.e. from which point onward the (de)activation conditions have to be considered[1]), the repair condition (R) describing the actions to undo the norm violation, and splits the aim into its verb and object (also called oBject by [142]), based on the work by [90, 120, 142, 144, 154]. Furthermore, to our ADICDlRO framework, we add the elements of *pro- and demoted values*, meaning that norms promote and demote values [81]. Finally for the agent to reason through the perceived consequences of the norm, we add the information of *negatively affected needs* to our framework. This is important, as it allows the agent to see if any need which is currently not satisfied is negatively affected by the norm, and thus the agent might feel an impetus to act upon that norm, such as circumventing the norm or violating [81] the norm. The negatively impacted needs, the promoted values, and the demoted values are unordered lists.

We acknowledge that norms can also positively impact needs, as is the case with values, e.g. social distancing positively impacts the safety need of a person. However, this is not relevant for our work at this point. With our needs model, we assume that people already strive for a balanced state, and as such would engage in social distancing behavior even without the social distancing norm being active, as it would satisfy their safety need. In this sense, the norm is focusing on a behavior that the person is already doing, and has therefore no potential impact, as the person does not have to change their behavior, as they already adhere to the norm. In our work, we focus on the potential negative consequences of norms, as negative impact is more likely to motivate us to change our behavior. While there is also the possibility of positive enforcements via rewards, e.g. a policy could be that a person receives a monetary reward for recycling,

[1]Inspired by how the law takes affect in the real world

the question arises if this is enough to change the behavior of a person, given that their financial needs are already satisfied, and thus the extra reward for recycling would not add anything for them. This is a complex question to answer, and thus out of the scope for this research and left for future work.

Now, assuming a total set of $\mathcal{NO} = \{no_i, \ldots, no_n\}$ norms in the simulation, we can define each element of our extended ADICDlRO framework, and thus a norm $no_i \in \mathcal{NO}$ as follows:

A: Attribute, the agent(s) that have to adhere to the norm. Formally: $A^{no_i} \subseteq \mathcal{A}$, with $\mathcal{A} = \{a_1, \ldots, a_n\}$ the total set of agents in the simulation.

D: Deontic, the deontic operator of the norm. Formally, $D^{no_i} \in \mathcal{D}$, with $\mathcal{D} = \{Obligation, Prohibition, Permission\}$ the typical deontic operators used for norm specifications.

I: aIm, the targeted behavior and object of the norm. We split this into I_{Verb} and I_{Object}, also done by [144]. The I_{Verb} specifies the targeted action(s) that the agents specified in the *Attribute* have to adhere to. Formally, $I^{no_i}_{Verb} = act \in \bigcap_{A^{no_i}} \mathcal{ACT}$. The norm can only target actions that all agents that must adhere to the norm can perform (with $\mathcal{ACT}$ denoting the set of all actions in the in the simulation, and $\bigcap_{A^{no_i}} \mathcal{ACT}$ denoting the set of all actions that the agents responsible for adhering to the norm have in common). Given that we have the targeted action, we can now define the targeted object of the norm, I_{Object}. Since the object(s) targeted by the norm can only be object(s) by the targeted action, we formalize the object of the norm, I_{Object} as follows: $I^{no_i}_{Object} \in O^{I^{no_i}_{Verb}} \subseteq O$, with $O = \{o_1, \ldots, o_n\}$ the total set of objects in the simulation and $O^{I^{no_i}_{Verb}}$ the set of objects required for the targeted action.

Deonitc + aIm: Together we can combine the Deontic and aIm of the norm now to identify the fulfillment ($fullC^{no_i}$) and violation ($vioC^{no_i}$) conditions of the norm. Formally, $DeoaIm^{no_i} = \langle fullC^{no_i}; vioC^{no_i} \rangle = \langle 2^S; 2^S \rangle$.

C: Condition, the activation ($actC^{no_i}$) and ($deactC^{no_i}$) condition of the norm. Formally, $C^{no_i} = \langle actC^{no_i}; deactC^{no_i} \rangle = \langle 2^S; 2^S \rangle$.

Dl: Deadline, when the norm takes effect in the simulation. This is similar to laws in the real world and when they take effect. Formally, $Dl^{no_i} = 2^S$.

R: Repair. The repair condition defines the action to be done to undo the norm violation. Formally, the repair condition of a norm is an ordered list of $R^{no_i} = act \in \bigcap_{A^{no_i}} \mathcal{ACT}$.

O: Or else, the punishment for violating this norm. While there can be many forms of punishment, such as actions or other norms now being active, etc., we decided that for simplicity, we only consider physical actions. The reason is not only that this is enough for showing the advantages for the agents to reason through the consequences of a norm, but also that many punishments in real life can be considered fines or other forms of monetary punishments for example. Furthermore, we consider the punishment to be done instantly. This is to avoid further reasoning about the urgency of the punishment and when it is done. Even though, this in itself can become a temporary norm, namely an obligation to adhere to the punishment, we leave this for future work, as it is not only not relevant for bringing our point across, but also highly complex, as each of these temporary obligation also require further punishments given that the temporary obligation is violated, for example the initial fine is not payed by the agent. Based on this reasoning, we formally define the punish (O) as: $O^{no_i} \in \bigcap_{A^{no_i}} \mathcal{ACT}$.

promValues: promValues is the list of the promoted values by the norm, and defined as follows: $promValues_{no_i} = [v_1, ..., v_n], with\ v_i \in \mathcal{V}\ \forall v_i \in promValues^{no_i}, whereby\ |promValues^{no_i}| \leq |\mathcal{V}|\ holds$

demValues demValues is the list of the demoted values by the norm, and defined as follows: $demValues^{no_i} = [v_1, ..., v_n], with\ v_i \in \mathcal{V}\ \forall v_i \in demValues^{no_i}, whereby\ |demValues^{no_i}| \leq |\mathcal{V}|\ holds$

affectedNeeds affectedNeeds is the list of the negatively affected needs by the norm. It is important to note here that a negatively affected need can be part of the needs of any perspective present in the simulation, and thus we have to look at the union of all needs present in the simulation. Formally, this can be defined as follows: $affectedNeeds^{no_i} = [ne_1, ..., ne_n], with\ ne_i \in \bigcup_{p \in \mathcal{P}} \mathcal{NE}_p\ \forall n_i \in affectedNeeds^{no_i}, whereby\ |affectedNeeds^{no_i}| \leq |\bigcup_{p \in \mathcal{P}} \mathcal{NE}_p|\ holds$

3.5 Agent Deliberation

With the formalisms in place, we can now conceptualize our proposed Perspective-Based Agent Deliberation Architecture (PBADA). A top level overview of our proposed architecture can be found in Figure 3.4. It is important to note here that the environment has an action (deliberation) level with the deliberation in brackets as it does perform some action, such as updating time, but does not actively deliberate about it.

We can see in this figure that norms come into play at all levels of the agent's deliberation process. Furthermore, we can already see here the potential different deliberation paths that the agent can take:

1. The simple deliberation level (blue background): The agent tries first to find a typical (usual) action, if the situation is known
2. The medium complex deliberation level (beige background): The agent needs to adapt the usual action
3. The complex deliberation level (green background): The agent needs to deliberate from scratch about an unknown situation. This can be due to the norms or the physical or social situation that the agent is in.

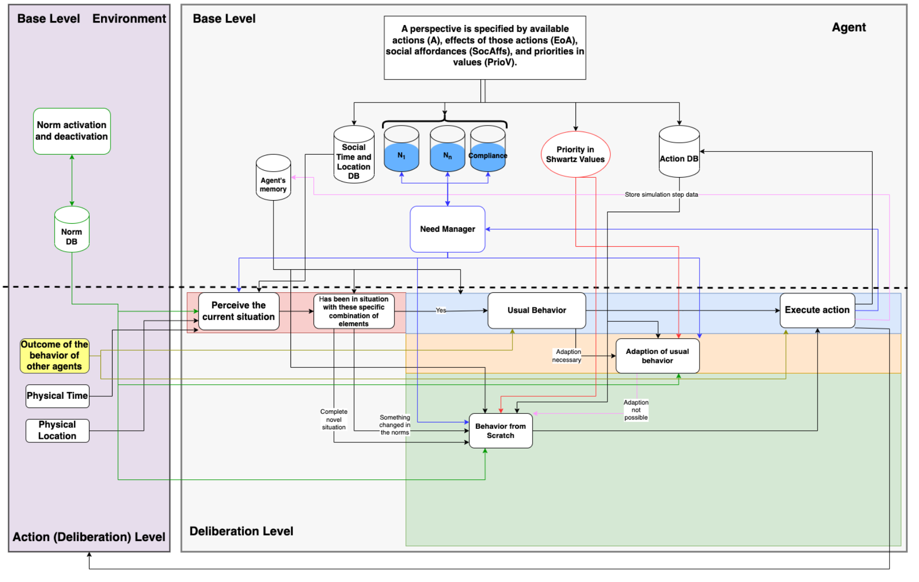


Figure 3.4: Top Level view on the PBADA architecture with the gray background being insight the agent and purple background representing the elements of the environment. The dashed horizontal line separates the base level and the deliberation level.

To show this in more detail, we are going to expand upon Figure 3.4, and present a more detailed overview of the agent deliberation architecture in Figure 3.5, where we show what is happening on each level of the agent's deliberation.

Figures 3.4 and 3.5 show that we can distinguish two levels in our proposed Perspective-Based Agent Deliberation Architecture (PBADA): the base level (above the dashed line) and the action deliberation level (below the dashed line). Note that wee added the arrows for elements, such as the norms or needs everywhere were they come into play (the colors for them are just for visual separation purposes). We could have also just added them at the beginning when the situation is perceived, but it is important to highlight and show that they come into play at various stages of our (the agent's) deliberation process. Furthermore, everything within the gray background is inside the agent, and the remaining parts are outside of the agent and part of the environment (purple background). Finally, the elements of the architecture are presented atomically to really show what is happening at each step and how the decision is coming about.

This idea of the separation between the base level and the deliberation level is based on the theory of metacognition by Nelson & Narens [115]. In their work, the authors describe two levels of cognition, the object-level, and the meta-level. On the object-level, the thinking in itself happens. We relate this to the deliberation level, where the decision-making about the next action takes place. On the meta level, the thinking about thinking is taking place. This is where the monitoring of the thinking takes place. We relate the meta-level to the base level of our deliberation architecture.

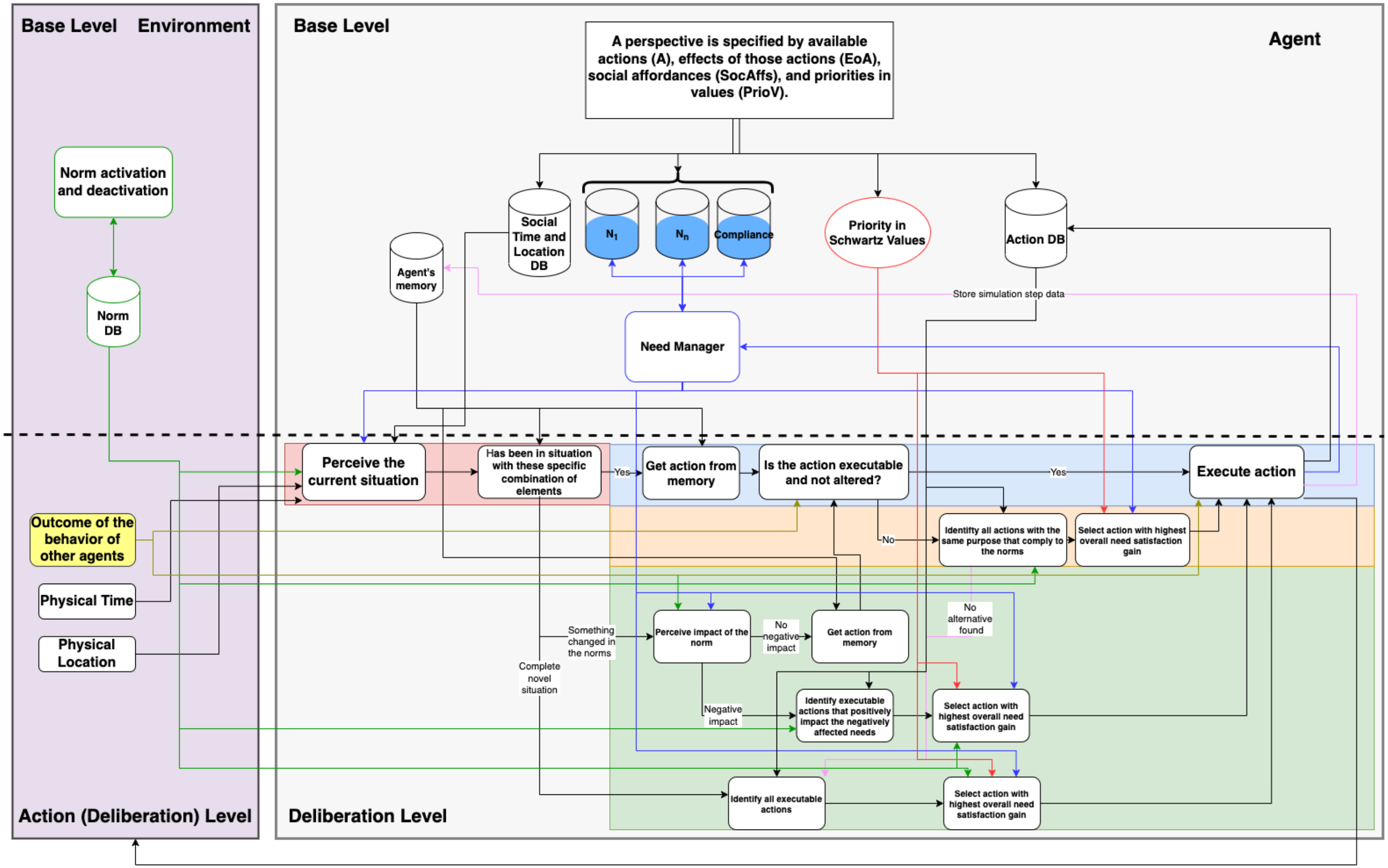


Figure 3.5: PBADA architecture with the gray background being insight the agent and purple background representing the elements of the environment outside of the agent. The dashed horizontal line separates the base level and the deliberation level.

We sketched the relation the between the meta-level and object-level together with our proposed PBADA agent deliberation architecture in Figure 3.6.

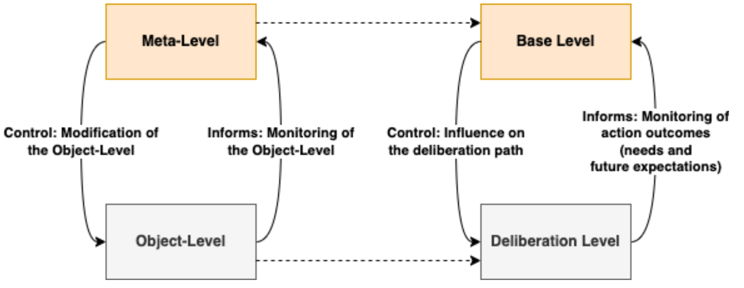


Figure 3.6: Metacognition levels and information flow adapted from the theory of metacognition by Nelson & Narens [115] on the left, and connection of our proposed PBADA agent deliberation architecture on the right. The shared colors and dashed lines are indication the mapping of the levels between each other. The meaning of the control and informs relationships are mapped from [115] on the left to our PBADA architecture on the right.

In the theory of metacognition, these two levels are connected via the processes of monitoring and control [115]. Control means that information flows from the meta-level to the object-level where "[...]meta-level *modifies* the object-level. In particular, the information flowing from the meta-level to the object-level either changes the state of the object-level process or changes the object-level process itself. This produces some kind of action at the object-level, which could be (1) to initiate an action [...]" [115, p.127]. We can relate this to our proposed agent-deliberation architecture, as the base-level (containing e.g. the current need satisfaction and the currently active norms), is affecting the perception of the agent, and thus decides on its further deliberation (and action selection).

Monitoring in the metacognition theory refers to "[...]that the meta-level is *informed* by the object-level. This changes the state of the meta-level's model of the situation, including "no change in state" [...]" [115, p.127]. This idea works for our proposed agent deliberation architecture, as the actions performed by the agent are satisfying their needs, and since not all of their needs affected by one action (e.g. working does not affect the pleasure need one gets from eating nice food), the no change in state mentioned by the authors is covered. Additionally, since need satisfaction by the actions can change, the agents memory is affected and updated by the provided information, and the norms get (de)activated by the norm manager based on the (information of) the simulation state (with the norm manager being in the base level of the environment). The needs further enforce the idea that the base level can be related to the meta-level by Nelson & Narens [115]. It is assumed that the meta-level contains some goal state that the person has [29, 114, 136]. While our proposed agent deliberation architecture does not have explicit goals (in the sense of reachable states), it has needs which the agent aims to satisfy, which can be related to this idea.

One thing missing in the metacognition theory by Nelson & Narens [115] is that the base-level (the meta-level) is monitoring and updating itself. We want to reflect this self-monitoring mechanism, as the needs are depleting overtime and becoming more salient. This means that something (need depletion) is happening in the base level (meta-level) without interference from the deliberation level (object-level).

3.5.1 The Base Level

Before the agent can make any decision, the agent needs to update their internal state and the world needs to be updated as well. We, as humans, do not just jump from action to action. We need a basis for our decision-making process. In this sense, this part of the deliberation architecture can be viewed as the meta-level part of the theory of metacognition by Nelson & Narens [115], as it influences the agent's reasoning about which action to take. We call this the *base level* of the architecture, as it provides the basis for the decision-making process of the agent. This can be found in Figure 3.7.

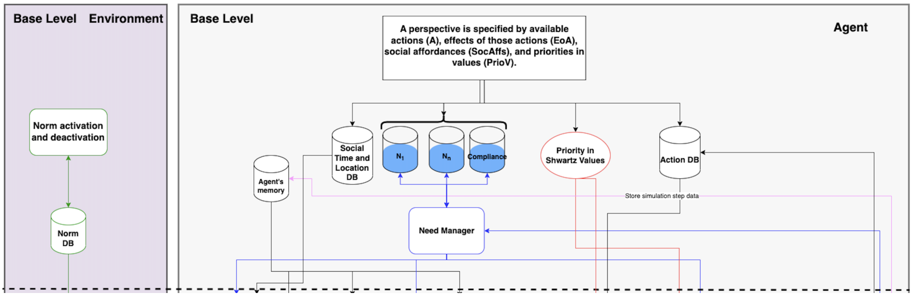


Figure 3.7: Agent deliberation architecture base level, as taken from Figure 3.5

The most important element is the needs. They are managed by the need manager. The need manager is responsible for updating the satisfaction of each need, i.e. the current fill level of the respective watertank, and calculating its urgency, given the depletion of the specific need, and the performed action. Algorithmically, this can be described as follows, see Algorithm 3.5.1 (the satisfaction update based on the executed action will be described later in the deliberation process).

The next element is the priorities in Schwartz [138] values (since we use them). They agent needs them to resolve conflicts in the form of equal perceived need satisfaction gains, i.e. use them as guidance which states are preferred. For example when two actions are providing the same perceived need satisfaction gain, the action that aligns more with the values of the agent is chosen. When two norms are in conflict with each other and only one can be adhered to and the perceived need satisfaction is the same for violating either of them (given the respective actions), the action that aligns ore with agents values is chosen. Furthermore, they are static over the course of the simulation.

To know which actions the agent can perform and reason about, the base level also contains the action database. This is where the agent can look at which actions to chose from. Having this database allows the agent to remember the current need satisfaction

Algorithm 3.5.1 Need satisfaction and urgency update

```
for each need n in 𝒩ℰ do
    n.currentFillLevel ← n.currentFillLevel −= n.depletion
    if n.currentFillLevel ≤ 0 then
        n.currentFillLevel ← 0
    end if
    n.urgency ← n.threshold −= n.currentFillLevel
    if n.urgency ≤ 0 then                          ▷ The need is satisfied
        n.urgency ← 0
    end if
end for
```

gain that the action provides, i.e. how attractive the action is. For example, the agent has now knowledge about how much pleasure to expect from going to the restaurant. This expectation can then also be updated over the course of the simulation, based on the behavior of other agents. For example cheaper ingredients used by the restaurant owner results in less expected pleasure for going to the restaurant. This means that the action database enables a form of memory for the agent to react to the behavior of other agents.

As we established earlier, the actions of an agent are tightly connected to the social affordances. This means that for the agent to be able to reason about which action to take, they need to have knowledge about what the current physical time and location is meaning for them, e.g. if it is currently working time or not. This information is stored in the social time and location database.

Finally, for the agent to be able to learn which action is preferred in which situation and to enable a faster habitual behavior, the agent needs a memory. This is for them to remember what to do in a particular situation. The agent's memory is storing the situations that the agent has been in over the course of the simulation. The stored situation contains the timestep, the physical and social location and time, the satisfaction of the agent's need at that point in time, the current active norms at that point in time, and the action that was executed at that particular timestep.

The environment monitoring its inner (base level) state means for the us the (de)activation of norms. This could either be if a certain time in the simulation passed by, e.g. two weeks, or another external event happened, e.g. the percentage of infected people for COVID models. This part is done by the norm (de)activation handler. The handler is accessing the norm database (holding the $\mathcal{NO} = \{n_1, ..., n_n\}$ norms which exist in the simulation) to see if the (de)activation condition is met for any norm in the current world state $w \in \mathcal{W}$, with $\mathcal{W} = \{w_1, ..., w_n\}$ consisting of all possible world states within the simulation. The (de)activation is as follows can be found in Algorithm 3.5.2 and Algorithm 3.5.3 respectively, with $N_{active} \subseteq \mathcal{NO}$ describing the current set of active norms and $N_{deactive} \subseteq \mathcal{NO}$ the current set of de-active norms.

Algorithm 3.5.2 Norm Deactivation

> **for each** norm n in N_{active} **do**
>   **if** $n.deactivationCondition \subseteq w$ **then** n.deactivate();
>   **end if**
> **end for**

Algorithm 3.5.3 Norm Activation

> $N_{deactive} = N \backslash N_{active}$
> **for each** norm n in $N_{deactive}$ **do**
>   **if** $n.activationCondition \subseteq w$ **then** n.activate();
>   **end if**
> **end for**

3.5.2 Action Deliberation Level

Now we can move from the meta-level to the object-level to stay in the terms of Nelson and Narens [115]. We call this the *agent deliberation level*, because the agent deliberates about their course of action and is interacting now with the objects in the simulation. This is the timestep-to-timestep business[1] of the agent, meaning that this process happens at every timestep in the simulation. The deliberation level of our proposed PBADA agent deliberation architecture can be found in Figure 3.8.

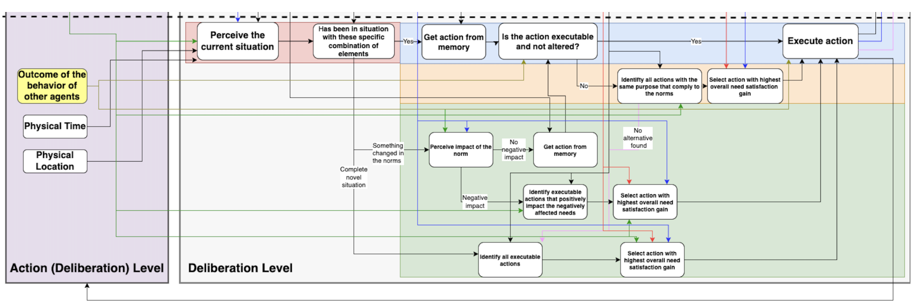


Figure 3.8: Agent deliberation architecture deliberation level, as taken from Figure 3.5

The connecting element between the base level and the decision-making process is the context part of the agent deliberation architecture highlighted by the red background in Figure 3.8. Context detection and checking of context familiarity (situational and normative) is part of the deliberation level as it is an active process by the agent taken at each timestep to determine the course of action at this particular point in time. Consequently, it is part of the deliberation process of the agent. The context detection is also the starting point of the deliberation process for the agent.

After the context familiarity has been determined we can divide the further

[1]Taken from the saying of the day-to-day business of a person

decision-making of the agent into the following three different layers of complexity. Each of these layers result in a different decision-making process of the agent not only in terms of the elements used, but also the time required to reach a decision:

1. The habitual (simple) deliberation level (blue background) allowing for a fast deliberation and habitual behavior choices. This happens when the situation and normative context is known to the agent, and they know what to do in a specific situation.
2. The medium complex deliberation level (beige background) for finding alternative actions, in case the habitual action changed in terms of expected need satisfaction or is not executable anymore.
3. The complex deliberation level (green background) is used to deliberate about an unknown situation. This can either be if something changed in the norms, e.g. a new norm became active, or a novel situation in terms of the most urgent need or time or location, e.g. the first time for the agent that it is Monday morning.

In the following sections, we will explain the elements of the deliberation level in detail. Since the agent starts with determining what level of deliberation is appropriate in the current situation, we start by explaining the context detection element. Then we describe how the agent is determining the context similarity. Afterwards, we explain the different deliberation levels in the following order: first the simple deliberation level, then the medium complex deliberation level, and finally, the complex deliberation level.

3.5.3 Context Detection

The agent first determines their current situation to detect their context. This means for the agent not only to be aware of their physical world, but also their social world, their needs (current need satisfaction), and the current active norms. All these elements are used by the agent to decide which deliberation path to take, and thus agents need to be aware of them. The needs are important, as they are the main drivers of behavior and determine which actions are the most attractive ones. This attractiveness is mediated by the current active norms, as some actions might conflict with norms, and the agent wants to adhere to the norms to satisfy their need of compliance (need to adhere to norms). Furthermore, the time and location (physical and social) determine the availability of actions. Formally, this results in the following tuple:

$$\text{currentSituation} = \langle \text{tick}, \text{physT}, \text{socT}, \text{physL}, \text{socL}, \mathcal{NE}, N_{active} \rangle \tag{3.7}$$

with: tick = current timestep, physT = current physical time, socT = current social time, physL = current physical location, socL = social location, the needs of the agent ($\mathcal{NE}$), and N_{active} = the currently active norms.

An example based on a restaurant owner could look as follows: currentSituation = ⟨ 3 (tick), Monday evening (physT), working time (socT), restaurant (physL), working place (soc.L), (financialstability, security, compliance) (needs of the agent), size-based restriction (active norm)⟩

3.5.4 Context Familiarity

Given that the agent is aware of their context, they assess whether they recognize the situation that they are currently in. Recognizing the situation means that the agent does know what to do and moves forward with the habitual habitual deliberation. If the agent does not recognize the context, they move to the complex deliberation level, as the situation (to some extend) is not known to the situation. To decide to which extend the situation is not known, the agent looks at the *situational context*, and the *normative context*. These are both important, because before the agent can make any decision about norms, they have to understand if the current situation is known to them. This very similar to the real world. Before we decide and think about any norms, we firstly orientate ourselves in the world, e.g. where we are or what time it is, before we think about any norms. Consequently, we live and decide in a multi-layered context: the situational layer, and the normative layer.

So we can say, that this is a crucial part in the deliberation process, as it determines how the agent is going to proceed in their decision making. To make this decision, the agent is required to access and query their memory to see if they have been in this particular situation before.

3.5.4.1 Situational similarity

Not every situation is new for an agent (a person). They have been in situations before, and thus can remember and know what to do in those situations. For example, not every Monday morning during working time is new for the agent. They might have been at this time last week or the week before.

To determine if the agent has been in the situation before, they first look at the physical and social time and location. Then given the most dominant need, the agent assesses if they have been in such a situation before. If this is the case, the agent has a potential action to go forward, i.e. what they would normally do at this time in this location given that most urgent need. This is roughly shown in Algorithm 3.5.4. It is important note there that we can do these comparisons, because we are assuming a meaningful grouping of the physical times and physical locations together in bigger groups, where meaningful depends on the specific simulation. This grouping is happening in reality as well, see for example shift work which is grouping specific times into a meaningful group.

This is the point where we utilize the memory of the agent that stores all the states that the agent has been in. To account for recency bias[1], we go through the agent's memory backwards with the most recent situation added.

If the algorithm returns true, we have an action, we call this the *habitual action*, and can proceed to see if something changed with regards to the norms. Otherwise, the agent treats this as a complete novel situation for them (complex deliberation level), as they do not know what to do with this specific combination of time, location, and most urgent need.

[1]Recency bias means that recent events have a higher impact on our decision-making than events further in the past [49], meaning that we think and act first based on what happened recently rather than something further in the past.

Algorithm 3.5.4 Situational context familiarity. The agent checks whether it has been in this location at a the same moment of the day before with same most urgent need

```
contextFamiliar ← 𝔽
for each situation s in assessedSimulationStatesDB do
    ▹ assessedSimulationStatesDB represents the agent's memory, iteration through memory with most recent entry first
    if currentSituation.physT == s.physT & currentSituation.socT == s.socT & currentSituation.physL == s.physL & currentSituation.socL == s.socL then
        if currentSituation.mostUrgentNeed == s.mostUrgentNeed then
            ▹ most urgent need identified by querying over all needs
            contextFamiliar ← 𝕋
            return s
        end if
    end if
end for
```

3.5.4.2 Normative Context

When integrating norms into the reasoning process of the agent, it is important to consider them when assessing one's situation. It might be possible that one (agent) has been in the same situation before, e.g. Monday morning during working time and being at home with the most urgent need being compliance to contractual obligations (i.e. work). But a new norm might be active since last week, for example that work from home is now mandatory. The agent now needs a way to decide if this norm is relevant for them.

To decide which norms are relevant, we look closely at the object (o_i) used by the habitual action, and see if anything changed in the norms that have the same object. This means that the agent looks at all currently active norms where $o_i = I^{n_i}_{Object}$ holds, with o_i being the object of the action and $I^{n_i}_{Object}$ the object used by norm n_i. It is important to do this (especially for large simulations), because not all currently active norms might be relevant at this point for the current action. For example, if the agent wants to go to the restaurant, they do not have to look at the current active norms concerning a hospital.

If nothing changed (meaning that the same norms are active and those also have the same definition), the agent moves into the habitual and fast deliberation layer to see if the action can still be executed. Otherwise, when a new norm became active or something changed in the current active norms, e.g. the the size-based restriction changed such that only 25% of the space is allowed to be used the restaurant owner instead of 50%, the agent moves to the norm reaction part (in the complex deliberation level) to asses the potential impact of the new norm or change in the current active norms.

Of course, it is also possible that a norm got deactivated. However, the reaction to a norm deactivation is not part of our research focus and thus assumed that in a model using our agent deliberation architecture that the agent will go back to the behavior before the now deactivated norm was active, since it is stored in the memory of the agent. To integrate the reactions to norm deactivations properly, further research into habits is required to understand to what extend people actually revers their behavior when the norm that caused the initial behavior change gets deactivated. Some people might keep the new habitual action, others might revers to the action before the norm, and yet others might seek completely new actions[1]. For example: Some agents will keep going to the park instead of the restaurant, as they still assume that the restaurant owner is using cheaper ingredients (which was a reaction by the restaurant owner to the now deactive norm). Some restaurants would switch back to using higher quality ingredients but then also react to the loss of reputation by using cheaper ingredients and do something to get their lost guests back. Other restaurant owners might keep the lower quality ingredients and try to find other ways to attract customers. All of these potential reactions to norm deactivation are out of the scope for this thesis, and are part of our proposed future work.

When looking at norm violating actions, such as letting in more guests than allowed (given a size-based restriction norm that is limiting the available space in a restaurant), it does not make sense of course to keep that action after the size-based restriction norm. This is covered in our agent deliberation architecture in the part when the habitual action is not executable anymore. Since we check there for the pre-conditions of the action, one can define them in such a way that they require the norm to be active for the action to be executable. This ensures that actions depending on specific norms being active (such as letting more guests in than allowed), are only executable when the corresponding norm is active.

3.5.5 Is the action executable and not altered? - Habitual Deliberation Level

Now that the agent determined their context and identified if they have been in this context before, they are now following a deliberation path to determine their course of action. In this section, we explain the simple (habitual) deliberation level (blue background in Figure 3.8), and in the following sections, the medium complex deliberation level (beige background in Figure 3.8), and (most) complex deliberation level (green background in Figure 3.8).

If the agent has been in the situational and normative context before, they have an action now, their habitual action. However, they cannot blindly execute the action. The agent is not living in a vacuum, but rather in a world with other agents. This means that the agent has to check, if the habitual action is still executable or something changed the agent executed the action the last time, thereby enabling the agent to learn about potential changes to their actions (e.g. the a restaurant owner is using cheaper ingredients which is affecting the action of the agent to go to the restaurant). This can then also be the place to account for the changes in the agent (or the environment), e.g. being sick might render some actions not executable anymore.

[1]This also refers to norm internalization and the transformation of legal norms to social or moral norms, see e.g. [65, 101, 102] who showed that people in the Netherlands stuck to the previous speed limit even after it was raised

To see if the action is feasible, we formally check if:

$$actionToExecute.isFeasable() == \mathbb{T} \tag{3.8}$$

holds. In this function, all pre-conditions of an action are checked to ensure that they hold. For example: When the action is going to the restaurant, the pre-conditions might be that it is evening, and that the restaurant is open. Equation (3.8) only returns true (holds) if both conditions are met.

This example shows nicely how the interaction with other agents comes into play here. For example: If the restaurant owner would have closed their restaurant, then Equation (3.8) would return false instead of true, as the check if the restaurant is open would fail.

When Equation (3.8) does hold true, the agent checks if the expected need satisfaction gain of the action has been altered since the last time that this action was executed, for example by having a boolean flag to see if something changed or comparing each expected need satisfaction from the current version of the action to the previous one. This could be for example the case when the restaurant owner is using cheaper ingredients (as a reaction to a norm) and thus the guests expect to get less pleasure from the action to go to the restaurant. If nothing has changed, the agent continues to execute the action and conclude the habitual deliberation. Otherwise, the agent moves to finding an alternative action wit the same purposem i.e. the medium complex deliberation level.

3.5.6 Finding and alternative action - Medium Complex Deliberation

We reach this part of the deliberation process now when the agent is not able to execute their typical habitual action or if the expected need satisfaction gains of that action have been altered. Colloquially this could be formulated for example as follows: Normally we would go to restaurant x now, but since the owner had to close restaurant x, we have to find an alternative. Alternatively, if the agent learned that something changed in the expected need satisfaction gain for the action it could be formulated as follows: Yeah let's go to restaurant x now. Oh no wait, they use cheaper ingredients now and the food is no that good anymore, let's see what other options we have.

To now select an alternative action, we use the *purpose of the original action*, i.e. the social action, to decide at which potential actions the agents should look at. For example, going to the restaurant was chosen with the purpose to socialize with their friends, and thus the agent is now looking at alternatives to socialize, such as going to the park. While this requires some more complex reasoning by the agent, we still consider this part of the quicker part of the agent deliberation process, and not the 'full' complex deliberation. We look rather quickly for alternatives when our typical action is altered or not feasible anymore, rather than engaging in a very complex and deep deliberation process.

With the above considerations in mind, we can describe in detail now how the agent is going to find an alternative action, see Algorithm 3.5.5. The agent is gathering all actions that have the same purpose and do not violate any current active norm. The reason we do not look at actions violating norms is that we want to find a quick alternative if we cannot execute our habitual action, as in reality we also look for a quick alternative and not deliberate about violating norms or not. Rather we look at an alternative action, and if it is violating the norm, we look for the next one. Otherwise, we would always end up and engage in a full complex deliberation process

Afterwards the agent selects the action with the highest overall perceived need satisfaction gain, where the perceived satisfaction gain is calculated as the sum over the actual need satisfaction gain from an action times the urgency of the specific need. The reason for this calculation is to utilize the strength of our need-based approach. Since needs deplete over time, some actions are perceived as more attractive. For example: Sitting on the couch provides objectively the same rest need satisfaction gain. When the person was out the whole day and working the rest need is depleted and thus, sitting on the couch is subjecticvely perceived as very attractive. However, if the person was at home the whole day laying in bed, the rest need is satisfied, and sitting on the couch is subjectively perceived as not that attractive. This is why we make this calculation and sum up the multiplication of objective need satisfaction gain times the urgency of the need for all needs, as actions can satisfy multiple needs.

If two actions have the same perceived need satisfaction gain, the agent chooses the one that aligns the most with their values. Aligning most with ones values means that the action which is promoting a value that is more important than any other value of the other action, is considered aligning most with ones values. We do it in this way, as the preferences between the values are categorical and not numerical. This means that we do not determine to what extend a value is more important than another value, but rather simply that it is. As a consequence, we cannot do any calculations, such as action one promotes value x and value y resulting in z, and action two is promoting values a and b resulting in c and since z is larger than c, action one aligns more with the values of the agent. Rather we say, if action one promotes value x and x is more important than values a and b (promoted by action two), then action one aligns more with the values of the agent than action two.

If these actions also promote the same values, then the agent keeps the one they selected first, as they go through the potential actions in the list sequentially.

Finally, if no alternative action can be found, the agent treats this situation as a completely novel situation. The decision-making of the agent in a completely novel situation is described in Section 3.5.8.

3.5.7 Something changed in the norms - Most Complex Deliberation

Now we enter the most complex deliberation part. This means that the agent is deeply engaging with the situation rather than trying to find a quick solution as in the previous simple and medium complex deliberation levels. We are not at the stage where the most time is spent and the most cognitive resources are used. For this, we first look at the case when something changed in the norms. This means that the situational context is is familiar to the agent, e.g. Friday evening on a working day with the most

Algorithm 3.5.5 Find alternative action

```
potentialActions ← new List()
highestNeedSatisfaction ← 0
alternativeAction ← null
for each action in actionDB do
    if action.purpose == habitualAction.purpose then
        for each norm in N_active do
            if action.postCondition ∉ norm.violationCondition then
                potentialActions.Add(action)
            end if
        end for
    end if
end for
if potentialActions ≠ ∅ then
    for each action ∈ potentialActions do
        perceivedSatisfactionGain ← 0
        for each affectedNeed ∈ action.affectedNeeds do
            perceivedSatisfactionGain += affectedNeed × UrgencyOfTheNeed
        end for
        if perceivedSatisfactionGain > highestNeedSatisfaction then
            alternativeAction ← action
            highestNeedSatisfaction ← perceivedSatisfactionGain
        else if perceivedSatisfactionGain == highestNeedSatisfaction then
            for each value ∈ PRIOV do
                if value ∈ action.promValues then
                    alternativeAction ← action
                    break
                else if value ∈ alternativeAction.promValues then
                    break
                end if
            end for
        end if
    end for
    return alternativeAction
else
    go to Section 3.5.8
    ▷ If no potential alternative action exists, the agent treats the situation as a complete novel situation
end if
```

urgent need to socialize and eat with friends. But something changed in the normative context, e.g. a sized-based restriction norm which is limiting the space available in restaurants is active now that was not active previously when the agent was going out to socialize and eat with their friends at a restaurant.

The first step is to decide between reactive and proactive behavior. Reactive means that the agent is not changing their behavior (for now), like a ‘wait and see’ approach. Proactive means that the agent is preemptively doing something against the norm, as they feel threatened by it. Doing something against the norm can be to violate it or circumvent it.

We do this by looking at the needs of the agent, since our motivators determine which action is the most attractive one. For this interplay, we use the information of the norm which needs are negatively impacted by it, e.g. the restaurant size-based restriction norm is negatively impacting the financial stability need of the restaurant owner. The agent looks if those needs are currently satisfied. If so, they stay reactive and continue to execute the habitual action, as they do not perceive a negative impact by the norm. For example: If the restaurant owner has enough money (i.e. their financial stability need is satisfied), they do not perceived the size-based restriction norm as a thread, and thus continue their habitual action. This brings then the agent back to the fast deliberation process to see if they can continue their habitual action, and if so proceed to execute the action.

Algorithmically, we do the following, see Algorithm 3.5.6:

Algorithm 3.5.6 See if norm has negative impact

```
normHasNegativeImpact ← false
for each need n in norm.negativelyimpactedNeeds do
    if n.urgency > 0 then                      ▷ We see if any need is not satisfied
        normHasNegativeImpact ← true
        break
    end if
end for
if normHasNegativeImpact then
    Be proactive
else
    Be reactive
end if
```

If at least one of the needs that are negative impacted by the norm is not satisfied, then the agent perceives a negative impact and wants to proactively do something against the perceived negative impact. For example: If the restaurant owner does not have enough money (i.e. their financial stability need is not satisfied), they do perceive the size-based restriction norm as a thread, and thus want to do something against that.

This means that in the first step the agent needs to identify what they can do against those consequences, e.g. for the restaurant owners to potentially increase their income. To do this, we equip the action database of the agent with a dictionary for each need, consisting of the need as the key and the list of actions that can be performed to satisfy this need as the value. Formally, it can be seen as a mapping of needs to actions where the need is the input and the list of actions is the output: $needToActions[n_i] = [a_1, \ldots, a_n]$, with $n_i \in \mathcal{NE}$ the specific need, and $a_i \in \mathcal{ACT}$ the actions satisfying that need. To make this more clear we look at a restaurant example: The size-based restriction norm negatively impacts the financial stability need of the restaurant owner. To do something against that, the restaurant owner can now use this dictionary to look which actions counteract this negative impact. In other words, the restaurant owner can now look at the actions which positively impact their financial stability need. This could then look as follows: $needToActions[financialStability] = [useCheaperIngredients, letMoreGuestsInThanAllowed]$.

Since the agent knows which actions can be taken to counteract the perceived negative consequences of the norm, they can now start and deliberate about which action is the best one to chose. The first step is to filter for all actions that are executable right now. After that, the agent calculates for all executable actions the perceived need satisfaction gain, and checks if the action is kept after any potential norm violations (which we will explain in more detail in the following paragraphs). At the end, the action with highest overall perceived satisfaction gain, after any norm punishment deductions, is chosen by the agent and executed. Algorithmically, this can be sketched as follows: see Algorithm 3.5.7 for the identification of counteracting executable actions, and Algorithm 3.5.8 for selecting the action with the highest overall satisfaction gain.

The reason for doing it like this is as follows: The needs are the key motivations (i.e. the key drivers) in our decision-making process. They determine the attractiveness of an action. So before deciding about norm violations, we need to know if a norm violating action is even attractive to us. This is why we calculate the perceived need satisfaction gain. Perceived means the sum over the need satisfaction gain times the urgency for each need affected by the action (see here lines 5-7 in Algorithm 3.5.8). Now that we know how attractive the action is for the agent, we can look what would happen if we would go through with it. Here come the norms into play. Since we are considering norm violating actions as well, we have to see if the action is still attractive after the punishment gets deducted, i.e. we have to ask ourselves the question if the action is still attractive if we would get caught violating the norm (see the next paragraph). Finally, we use values to decide between two actions of equal attractiveness, as mentioned in the case of finding an alternative action,

The last part that we have to talk about here is the calculation of the, what we call, *violation score*. The basic idea behind this is that when a person wants to violate a norm, they compare the outcome of the action minus the punishment, to the benefit of following the norm, i.e. the saved punishment and compliance (the need to adhere to norms) gain minus the satisfaction loss, and if the perceived satisfaction gain is higher than they decide to keep the action. More colloquially, we can formulate it as follows: If we get more from executing the action than we lose from violating the norm, we keep the action. For example: If action *a* has a perceived satisfaction gain of *x* and the loss of the compliance need satisfaction (which is the one relevant for adhering to the norm) plus the punishment is *y*. Then what the person basically does is to look if *x - y > 0*. If this is the case, the person keeps action *a* and otherwise drops it and looks

Algorithm 3.5.7 Identify executable actions

```
potentialActions ← new List of Actions
executableActions ← new List of Actions
for each need ∈ norm.NegativelyEffectedNeeds do
    for each action ∈ needToActions[need] do
        if !potentialActions.Contains(action) then
            potentialActions.Add(action)
        end if
    end for
end for
for each action ∈ potentialActions do
    if action.isExectuable() then        ▷ Checking here if the pre-conditions hold
        executableActions.Add(action)
    end if
end for
return executableActions
```

for another one. It is important to note here that the perceived satisfaction gain must be strictly higher (>) than the punishment plus compliance lost. In case of an equal result, i.e. x - y = 0, we decided that the person will adhere to the norm. Of course, one could say now to look at the values in this case and if the values promoted by the action and more aligned with person than the values promoted of the norm, the person will violate the norm. However, we argue here that in general people want to follow norms when possible, and thus if the action is not clearly better than adhering to the norm, people will follow the norm.

To be able now to do these kinds of calculations and comparisons, we need to have a way to calculate the punishment in terms of its perceived impact on the needs of a person[1], we need a mapping from the exact punishment to its type of punishment which we can then use later for further calculation. The reason for this is that the decision-making of a person is based on their needs. The challenge is that the punishments for norm violations are formulated very explicitly, especially when it comes to legal norms[2]. For example with regards to a restaurant scenario: the punishment for letting in more guests than allowed might be a fine of 200 Euros. Now this 200 Euro fine has a different relative impact on the needs of the restaurant owners based on their current capital. Restaurant owners that have a lot of capital might not feel as strong of a impact compared to restaurants owners that do not have a lot of capital. Also, there is typically no need that this called 'fine'. This is were the aforementioned mapping comes into play.

This mapping represents a simple function f which take the punishment as the

[1]Here we use the same approach as for the perceived need satisfaction gain of actions, i.e. the sum over the punishments times the respective urgency of the associated needs

[2]our main focus in this thesis

Algorithm 3.5.8 Select action with the highest perceived need statisfaction gain

```
highestNeedSatisfaction ← 0
actionToExecute ← null
for each action ∈ executableActions do
    perceivedSatisfactionGain ← 0
    for each affectedNeed ∈ action.affectedNeeds do
        perceivedSatisfactionGain += affectedNeed × UrgencyOfTheNeed
    end for
    if action is violating a norm then
        Calculate violationScore in Algorithm 3.5.9
        Check for chance of getting caught in Algorithm 3.5.10
        if keep action from Algorithm 3.5.10 then
            perceivedSatisfactionGain −= vioaltionScore from Algorithm 3.5.9
        else
            Drop action
        end if
    end if
    if perceivedSatisfactionGain > highestNeedSatisfaction then
        actionToExecute ← action
        highestNeedSatisfaction ← perceivedSatisfactionGain
    else if perceivedSatisfactionGain == highestNeedSatisfaction then
        for each value ∈ PRIOV do
            if value ∈ action.promValues then
                actionToExecute ← action
                break
            else if value ∈ actionToExecute.promValues then
                break
            end if
        end for
    end if
end for
return actionToExecute
```

input and presents the type of punishment as the output, formally:

$$f\colon \text{punishment} \to \text{typeOfPunishment} \tag{3.9}$$

This can then look in our restaurant example like this: $f(\text{fine}) \mapsto \text{capital}$. The next step is to calculate the relative impact that the punishment has based on the the type of punishment. It is exactly this relative impact that is important and will be used for the impact on the needs of the agents. Since the needs are very specific for each model, we use the restaurant scenario here as an example: given that the fine is 200 Euros and the restaurant owner has a capital of 2000 Euros at the point of the violation, the relative impact (*relImpact*) is 10% (200/2000).

Now that we have the relative impact, we continue with our second mapping to connect the *typeOfPunishment* to the specific need, and the *relImpact* to the satisfaction loss of that need. Formally, we do the following (Equation (3.10)):

$$\begin{aligned} g\colon \text{typeOfPunishment} &\to \mathcal{NE} \\ \text{relImpact} &\mapsto [0,1] \end{aligned} \tag{3.10}$$

This is a categorical mapping[1] where the *relImpact* is mapped to a range of values. In our restaurant example this could look like this: $g(\text{capital}) \to \text{financial stability})$: $10\% \mapsto 0.2$. We want to note here that for the current version of the agent deliberation architecture we made this mapping absolute. For future research and iterations of the agent deliberation architecture, the mapping can be made more dependent on the values of the agent. In the current version, the modeler needs to make sure that a proper relation is maintained, i.e. if money is important to the agent, then a monetary punishment should have a higher impact in this mapping (lead to higher punishment costs). In future versions, this relation could be made explicit in the formula of the mapping.

Similar to the perceived need satisfaction gain, we have to calculate the *perceived punishment cost*. To do so, we have to calculate for each type of punishment, as well as for the compliance need the perceived impact (urgency times impact). This does not only allow us to provide the agent with a way of reasoning about the perceived impact of the punishment for violating the norm, but especially the compliance aspect allows the agent to take its history of complying to norms into account. This means that if the agent was recently violating norms, they are more likely to adhere to norms more often until they have replenished their compliance need again. Vice versa this also means that if they adhered to norms in the recent past, they might be more likely to violate a norm, as they 'feel' that they 'behave well'.

This is analogue to calculating the perceived satisfaction gain. To summarize everything algorithmically, we calculate the violation score as follows, see Algorithm 3.5.9, assuming the calculation score for norm $n \in \mathcal{NO}$. We want to remind the reader here that the punishment of violating the norm is the *Or Else* of our proposed *ADICDlRO* framework which is based on the *ADICO* grammar by Crawford & Ostrom [25]. We refer to it here as the punishment to make the function of this element of the norm clearer.

[1] We use a categorical mapping at various stages in this part, because people do not necessarily focus on the specific value but rather in what range it is.

Algorithm 3.5.9 Calculation of the violation score

```
violationScore ← 0
perceivedComplianceImpact ← n.complianceLoss × compliance.urgency
violationScore += perceivedComplianceImpact
for each punishment ∈ n.punishments do
    typesOfPunishment ← normDB.punishmentToType[punishment]
    ▷ Dictionry with relation of punishment (as key) to type of punishment, stored in norm DB
    for each type ∈ typesOfPunishment do
        punishedNeeds ← normDB.punishmenTypeToNeeds[type]
        ▷ Dictionry with relation of typ of punishment (as key) to needs, stored in norm DB
        for each need ∈ punishedNeeds do
            needImpact ← result of Equation (3.10)
            ▷ Get need impact from previously explained mapping functions
            perceivedneedImpact ← needImpact × need.urgency
            violationScore += perceivedneedImpact
        end for
    end for
end for
return violationScore
```

We want to mention here that while this looks big and intimidating with the three for each loops, it can be a rather fast process depending on the model, because the punishments and subsequent mappings can be only one to one relations, meaning that each for each loop is only executed once. Nonetheless, we wanted to show in Algorithm 3.5.9 that our approach is a general one and can be used for multiple punishments relating to multiple needs, i.e. many to many relations.

Now that we have the violation score, we can compare it to the perceived satisfaction gain, i.e. we can see if the perceived satisfaction gain is larger than the violation score. If that is the case, we can move on to the last part of deciding to violate the norm. That is to what extend the agent is likely caught to be violating the norm. It is important to consider this, because if the chance that the agent is getting caught violating the norm is lower, they might be more likely to violate it. To integrate this into the reasoning process of the agent, we are taking again a categorical[1] approach, meaning that if the difference between the perceived satisfaction gain and the violation score falls within a certain range, the agent is willing to violate the norm. It is important to note here that these ranges are dependent on the type of punishment and can thus vary. Algorithmically, this can be sketched as follows: see Algorithm 3.5.10

[1]Since people rather focus on the range than the exact value for their decision.

Algorithm 3.5.10 Check for the chance of getting caught violating the norm

```
difference ← perceivedSatisfactionGain − violationScore
if difference ≤ 0 then
    drop action
else
    for each range ∈ Ranges do                ▷ Ranges that the difference can be in
        if range.lower ≤ difference ≤ range.upper then
            if chanceRange.lower ≤ chanceOfGettingCaught ≤ chanceRange.upper then    ▷ Range for chance getting caught violating the norm
                keep action
                break
            else
                drop action
                break
            end if
        end if
    end for
end if
```

3.5.8 Complete Novel Situation - Complex Deliberation

The last situation that we need to discuss before the action execution, is the case when the agent finds themselves in a complete novel situation. This can happen, when the agent is in a novel combination of time, location and most urgent need, for example on a Monday morning during working time. Then agent deliberates over all actions that are executable in this situation. Overall the deliberation is similar to the case when the agent deliberates about actions to counteract a norm. The only difference is that instead of only looking at actions which can counteract the perceived negative consequences of a norm, the agent is looking at all possible actions. This means that the list of potential actions in Algorithm 3.5.7 is filled with all actions of the agents which are then checked for being executable, and not only with actions that counteract a norm. The rest of the deliberation is the same, i.e. the perceived satisfaction gain and the check for the potential violation of a norm. Thus we omit showing here to not clutter the text and refer to the previous section and Algorithms 3.5.8 to 3.5.10.

The other reason that the agent can be here is if no alternative action is found in the medium complex deliberation level. Coming here because of this reason is rather ‘just a fail-safe’, as in most cases an alternative is available. Of course, when being here for this reason, the original habitual action is not part of the action set that the agent deliberates about.

3.5.9 Action Execution

The final part of the deliberation architecture is concerned with the execution of the action. When the agent reaches this point, they have determined the action which they want to execute in this timestep. The action execution process itself can be spitted into three different steps. The first one is to execute the action itself, i.e. apply the state transformation from the pre-condition to the post-condition of the action. It is important to note here that actions cannot fail and potential coordination with other agents is assumed to be implicitly done. For example, if the action to be executed is to go to the restaurant to meet one's friends, then it is assumed that the friends are there and no further coordination needs to happen.

After this is done, the agent evaluates the outcome of the action. Here the agent compares their expected need satisfaction gain of executing the action, to the actual need satisfaction gained after executing the action. The expected satisfaction gain is the one that is stored in the action information in the action database of the agent at the beginning of the timestep (the objective need satisfaction gain). It is important to note here that this is different to the perceived satisfaction gain used in calculations previously, because there we multiplied the expected satisfaction gain with the urgency of the need (the subjective need satisfaction gain). The actual satisfaction gain is the one that the agent will experience in this current timestep after executing the action.

This means that we are here not only accounting for a potential mismatch in the agent's perception of the action, but also allow the influence of the behavior of other agents. In a restaurant scenario example it could be the case that the restaurant owner decided to use cheaper ingredients in their dishes (to combat the potential loss of income) since the last time that the guest was there. For the guests this would then mean that they get less pleasure or enjoyment from going to the restaurant, as the food does not have the same quality for them anymore. If this change in ingredients has happened between the last time and the current visit of the guest, the guest would expect the old pleasure satisfaction gain from going to the restaurant and make their decision based on that. Now, that they 'learned' and 'experienced' the new (lower) quality of ingredients, they update their expectations (the satisfaction gain of the restaurant action) for the next time. Furthermore, the needs are updated with the newly learned satisfaction gain, and not the old (and now outdated) one. This evaluation is crucial, as the actions of agents do not happen in a vacuum but within the world and potentially affect other agents as well, in our restaurant example the quality of ingredients. This is important when aiming for realistic simulations for policymaker support. We showed this importance in our ASSOCC project [37], where the agents decided to socialize with their friends based on how many they expected to show up (i.e. their expected belonging need satisfaction gain), and then evaluated their actual belonging need satisfaction gain based on how many friends actually showed up [37, 88].

Finally, after the outcomes of the action have been evaluated, and the expected need satisfaction gains of the executed action are updated for the action, the agent is going to update their needs and storing the current simulation step in their memory. Note here that the addition of the current simulation step to the agent's memory needs to happen before the need update as otherwise the internal situation of the agent is not the same anymore for which the action was originally selected. When storing the data, the agent is adding to the original tuple of the current context, gained from Equation (3.7), the action that has just been executed. Furthermore, the first element of the tuple (the current timestep) is used as the key of the dictionary that is representing the memory of the agent. Additionally, we want to note that we use the term *satisfaction gain* when talking about the effects of an action on a need. But this does not mean that the gain is only positive. It can also be negative, as going out to eat requires spending money, which has a negative effect on the financial need the agent. Algorithmically, this process can be described as follows: see Algorithm 3.5.11

Algorithm 3.5.11 Storage of the current simulation step in the agent's memory and need update

```
currentSituation.Add(executedAction)                       ▷ Adding the executed action
timestep ← currentSituation.tick                ▷ Tick is the name of the time unit
currentSituation.Remove(tick)
assessedSimulationStatesDB.Add(timestep,currentSituation)          ▷ The agent's memory
for each affectedNeed ∈ executedAction.affectedNeeds do
    affectedNeed.currentFillLevel − affectedNeed.satisfactionGain
    ▷ For each affected need, add the satisfaction gain to its current satisfaction level
end for
```

This is the end of the deliberation process of the agent. Once the deliberation process has been executed by all agents, the overall simulation time moves on and the next simulation steps starts.

3.6 Conclusion

In this chapter, we have theoretically and conceptually described our proposed Perspective-Based Agent Deliberation Architecture (PBADA) allowing for subjective and situational engagement with norms. To allow for this subjective reasoning, we introduced and formalized the concept of perspectives that an agent can have (currently one fixed perspective). Furthermore, we provided a formalism of norms (the ADICDlRO framework) necessary for agents to utilize the norm in their deliberation process, as well as making the norm an explicit object in the PBADA architecture. Additionally, we have also integrated different levels (base level, deliberation levels) in our PBADA architecture to account for the different processes going on and the different complexities within the deliberation level, thereby allowing the PBADA architecture to be scalable.

The next chapter implements our PBADA deliberation architecture in a model to verify an validate our PBADA deliberation architecture. However, before we are going to do that, we want to discuss some limitation and avenues for future work which resulted from the work presented in this chapter.

3.6.1 Future Work

The work presented in this chapter reflects a foundation for a new generation of norm-realistic social simulations for policymaker support. We envision many relevant future steps, e.g. the coordination with other agents, and modeling of a social network explicitly.

Furthermore, there are many other elements than be part of future versions of our PBADA architecture such as goals, plans, and plan patterns [78]. Goals are states that the agent wants to achieve, but might be necessarily available in one step, i.e. with one action [78, 81], this is also where planning comes into play [78, 81]. While goals can be an element between needs and actions, i.e. actions achieve goals, and goals satisfy needs[78], there is still a lot of research to be done to properly integrate goals into the reasoning process of the agent.

For example: Let's assume it is Monday, and we want to meet our friends on Friday evening. This means that we form the goal to go to the restaurant of Friday evening. Then we would make a plan to achieve that goal, which would include actions such as reserving a table or finishing work at a certain time. Multiple problems can arise in this situation however. One is that of goal abandonment. For example, meeting our friends during the week might partially satisfy our need to socialize with them, reducing its importance and might make another goal for Friday evening more important (attractive), due to other needs depleting over the week. Another issue is managing multiple goals (that do not conflict with each other), and plans (that could potentially create conflicts with other goals or plans=. For instance, while planning to meet friends on Friday evening, we might also have work-related goals that we want to achieve, such as meeting a deadline of a paper submission on Friday afternoon. These two goals are not conflicting with each other as they are targeting two different times of the day. The challenge is now potential plan interference. It might be possible that planning for meeting friends in the evening could potentially interfere with the plan to the paper submission deadline, as time reserving a restaurant for meeting with friends is spent the same time that the agent allocated for working on the paper. Balancing these goals and plans simultaneously is complex and warrants further research before being fully integrated into an agent's deliberation process.

Another avenue for future research is the integration of more social rules into our proposed PBADA agent deliberation architecture. While our focus was on legal norms, there are other types of norms, such as social norms [124]. Furthermore, [102, 101] showed the relation between different social rules, such as norms, social practices, and conventions. Legal norms can impact these other social rules, and then it can depend to what extend the legal norms interfered with them, which can then determine what the likelihood is of them being followed. This also goes for habits, as we have already touched upon earlier in this chapter.

In addition, we assumed that the punishment is payed automatically when the agent is getting caught violating the norm. However, this is not always the case in reality, and sometimes there is a certain time frame in which the punishment can be paid. For example, the fine for illegal parking does not have to be paid immediately. There is usually some time frame in which the fine can be paid. When we integrate planning and goals into our proposed agent deliberation architecture, we can also add this delayed punishment payment, as then resolving the punishment can become a goal of its own [81]. These can also have multiple extensions to it, as for example missing the deadline for paying the fine for illegal parking usually results in an additional fine to be paid which can then again become a new goal.

Also, we did not focus on norm deactivation and reactions to them, as discussed in the normative context section of this chapter. This also presents an avenue for future research to investigate how people react to norm deactivation, and how to integrate them properly into the deliberation process of the agent.

Finally, the focus in this thesis was on the modifiability of norms, i.e. altering existing norms or adding new ones. In future research, it is also interesting to investigate the possibility of modifying and adding new perspectives to the simulation. This is, however, even more complex than modifying existing norms or adding new norms. One reason is that more elements need to be specified by the user and then also their impact on the existing elements has to be analyzed further. This is especially relevant for existing norms. When adding a new perspective, it might be possible that new needs are introduced into the simulation. When this is the case, the existing norms have to be analyzed to see which ones potentially negatively impact those newly added needs, and to what extend, i.e. just a little or a lot. Further research on how to do this properly is required.

To summarize, the outlined potential areas for future research show that there is plenty of work still to be done. This highlights that our proposed PBADA agent deliberation architecture is only the first step and opens the door for a new generation of norm-realistic social simulations for policymaker support.

Chapter 4

Agent Deliberation Architecture in Action

4.1 Introduction

In this chapter, we instantiate our proposed PBADA agent deliberation architecture in a model. With this model, we will run various different scenarios to demonstrate and gather evidence [150] to show that models using our proposed agent-deliberation architecture are indeed showing norm-realistic behavior by being capable of dealing with different perspectives on norms (as the core of our PBADA architecture). This is to verify and validate our PBADA deliberation architecture through the model. As such, the model acts as a proxy for the agent deliberation architecture.

Verification and Validation (V&V) is important to ensure that the agent deliberation architecture is scientifically sound and meeting the requirements: including more behavioral realism through including different ways norms can influence behavior.

V&V also has practical importance. V&V is essential for establishing trust and credibility [91] in the agent deliberation architecture, which is crucial for policymakers. This credibility allows policymakers to confidently use a model to explore and assess potential policy impacts. Trust in a model stems not only from the V&V process, but also from the experience of interacting with the tool itself (discussed in the next chapter). Together, both V&V and the interaction tool support a model's reliability and user confidence.

Verification means that the model is behaving as it is supposed to without any bugs or artifacts [160]. In our case this means that the implementation is reflecting the underlying architecture correctly. This is reflected in Gilbert & Troitzsch [55] saying that the model works according to our (the modeler's) expectations. We will detail this in Section 4.5.

The validation of our PBADA architecture is not as straightforward as the classical notion of validation, because comparing the model results to reality [86, 91, 132] is not feasible. Our PBADA architecture can be used for various contexts. It is a *general agent deliberation architecture*. This means that we would need many models (to reflect a wide range of application domains to prove its validity) to instantiate our architecture with and then analyze them to show how the normative aspects work and that our deliberation architecture is indeed suitable to account for them.

To solve this challenge, we instantiate our PBADA architecture in one model which is representative for various other possible models, and see that the result are matching reality in a plausible (reasonable) way. To assess if this is the case, we use the systematic experimentation approach proposed by Kleijnen [86]. This approach implies systematic experimentation with the model. We do this by changing one factor at a time between different scenarios. For each scenario, we compare the results with reality in a plausible (intuitive,reasonable) way, i.e. according to the expectations (stylised facts [67]) that one can intuitively see in reality.

This process, allows us to incrementally add factors like norms or expand the agents' action space to evaluate our PBADA architecture systematically. The approach ensures the architecture can model agents' individual and situated interactions with norms—whether they comply, circumvent, or violate them. Additionally, by using trace validity[1] (as suggested by van den Hurk et al. [74]), we can explain the model outcomes by tracing them back to specific agent behaviors (i.e. trace validity), validating our PBADA architecture and showing its explanatory power.

4.2 The restaurant example

In this chapter, we use the scenario of restaurants operating under a size-based restriction on the number of guests allowed inside, inspired by COVID-19 regulations. We used this scenario in our previous work in [78, 81, 82], as it serves for a great example to show various norm responses from different groups. The norm is as follows: A restaurant owner must not have more than 50% of their total capacity of guests present concurrently.

This example is both simple and insightful for illustrating norm responses: it is easy to understand yet reveals complex dynamics when considering the varied responses of different groups. While many groups are indirectly affected, such as restaurant staff, suppliers, and the broader community, we focus here on the two directly impacted groups: restaurant owners and guests.

For restaurant owners, the restriction has a negative financial impact. Fixed costs (e.g. rent, heating, electricity, and insurance) stay the same, but their income decreases due to their limited amount of guests. To counteract this negative impact, restaurant owners may consider strategies to increase their income per guest, or violate the norm. For example, an owner might raise the prices per dish or use cheaper ingredients to cut costs, balancing this against the potential impact on customer satisfaction. Alternatively, they might allow in more guests than allowed, such as regular guests, or everyone, depending on financial pressures, the importance of certain guests, and perceived risks of getting caught violating the norm. Letting more guests in than

[1]Trace validity refers to the ability to explain the outcomes of a model based on the specific agent behaviors, i.e. the outcomes can be traced by to the behavior of the agents [74]

allowed has not only a financial impact but also a social one, such as the customer relations that might be important to some restaurant owners. This is to make sure that they have a good relationship with their regular guests by accommodating for them despite the norm.

Guests, on the other hand, are primarily affected by the social implications of the norm, as they go to the restaurant to enjoy eating and socializing with friends and family. In response to limited seating, guests may opt to reserve tables more often, adjust their eating times to avoid peak hours, or go to parks (or other public places). Not all guests will alter their behavior; those who feel assured of getting a spot may continue as usual.

We acknowledge that the real world is more complex. However this example allows for demonstrating the relevant complexities surrounding norms, and is thus sufficient for verifying and validating our agent deliberation architecture by covering essential behaviors: obeying the norm, obeying the norm and taking additional actions against the perceived negative consequences of the norm, and violating the norm.

4.3 The model

In this section, we describe the model that is used to instantiate our proposed PBADA agent deliberation architecture. The model is based on the restaurant example described in the previous section (Section 4.2).

The model reflects a world of agents that centers around going to work, resting at home, going out to eat (nice) food and socialize with friends and family (guests), and running a restaurant (restaurant owners). The model contains the following two perspectives on norms: guests and restaurant owners. These two perspectives will be modeled with our definition of a perspective on norms, see Definition 1 in the previous chapter.

Guests will be able to make a decision about their next action (such as work or rest or go to the restaurant) at every timestep[1] in the simulation (i.e. multiple times during the day), while restaurant owners will only be able to make a decision at the beginning of each week. This is because the restaurants can have good days and bad days during a week, and thus the restaurant owners are unlikely to immediately worry if at certain days no or only a few guests come. The agents for each perspective will be described in more detail in Section 4.3.1. There, we will provide a detailed explanation of the agents' needs, values, and actions, and what happens in every timestep in the simulation.

The interaction between the agents is based on the medium complex deliberation level of our PBADA architecture, recall Figure 3.5. The interaction is based on the needs and different actions that the agents have. Guests can react to the behavior of restaurant owners. But guests cannot react to the behavior of other guests, and restaurant owners cannot react to the behavior of other restaurant owners. Having this interaction between agents shows that reactions to norms do not happen in a vacuum, but also affect others. Furthermore, having one interaction is sufficient, as it is clear

[1]Each distinctive point in time in the simulation is called a timestep

that if this interaction works that further interactions also work. Hence, this would only clutter the model, and distract from the main purpose to verify and validate our proposed PBADA agent deliberation architecture. For this purpose it is sufficient to show that the elements of our architecture work and thus are covered in the model, which is the case with one interaction possibility, the potential reaction of the guests to the restaurant owners' behavior.

The model contains four locations: 1) homes where the guests live, 2) the restaurants where the guests can go and the restaurant owner lives and works, 3) the workplace of the guest agents, and 4) the parks. We omit the inclusion of traveling between the places, as it is not relevant for the purpose of our model. Consequently, the agents teleport to the different locations between the timesteps in the simulation.

We kept the model environment as minimal as possible, as the focus is on the agents, how they use our PBADA deliberation architecture, and to show that this results in norm-realistic behavior.

The norm used in this model is the size-based restriction norm described in the restaurant example.

We will explain the elements in the following sections in the following order in more detail. First we describe the agents, the guests, and restaurant owners in more detail. Then, we will explain the initialization of the model. Afterwards we provide a process overview of the model, and discuss the choice of our implementation platform.

4.3.1 Agent Make-UP

Concretely in the model, we have the guest and restaurant owner agents. Table 4.1 shows an overview of the main elements and processes present in the model and if they are instantiations of our agent deliberation architecture or just added because of the model. Instantiation of our agent deliberation architecture means that the elements in itself are present in our PBADA architecture, e.g. the needs of an agent, but are now filled with model specific values, such as pleasure. The guest characteristics are part of the agent deliberation architecture, as they affect the needs, such as varying thresholds, and are thus part of the instantiation of the architecture. The restaurant changing mechanic is a model specific addition, as it is not present in the architecture, but added in the model to increase the model specific behavioral realism of the guest agents. The environment refers to the simulation environment. These are elements that exist in the simulation model but are outside of the agent. After describing all the elements, Figure 4.1 in Section 4.3.3 provides a process diagram of what is going on in teach timestep in the model.

4.3.1.1 Guest Agent Make-Up

The key characteristics of the guests that are specified in this section are their needs, priorities in Schwartz values, actions, their characteristics (to introduce heterogeneity), and their social times and location. The guest needs can be found in Table 4.2. While in general the actual needs depend on the specific model itself, we took inspiration from our ASSOCC project and the needs modeled there [75] which in turn is inspired

Table 4.1: Elements (above the divider) and Processes (below the divider) of the agents in our model based on our PBADA agent deliberation architecture. The process of the guests for changing their restaurant is a specific addition for this model.

Elements and Processes	**Perspective and Environment**
Needs	Guests and Restaurant owners
Priorities in Values	Guests and Restaurant Owners
Actions	Guests and Restaurant Owners
Physical Time, Location and Social Time, Location Database	Guests and Restaurant Owners
Characteristics of agents (effect on the needs and decision-making process of the agents)	Perspective specific characteristics to have heterogeneous guests and restaurant owners
Decision-Making Process (base level, and deliberation level)	Guests and Restaurant Owners
Norm (de)activation	Environment
Simulation time update	Environment
Restaurant changing mechanic	Specific model addition for the guest agents

by Maslow's hierarchy of needs [94]. We used these needs as a basis and adopted them for this specific model. Similar to the ASSOCC project [75], we did not implement the strict hierarchy proposed by Maslow [94], because humans seek overall a balanced state, and can prioritize based on what is important to them.

The priorities in Schwartz values [138] of the guest agents are: $[Benevolence, Universalism, Security, Conformity, Tradition, Stimulation, Hedonism, Self-Direction, Power, Achievement]$. This priority in values of the guest agents ensure that in case of conflicting situations, the agents prioritize to go out to eat and socialize. This is important, as this is a key component in our model. In this way the action to go out to the restaurant would be chosen over staying at home, in case both provide the same perceived overall need satisfaction.

To reflect the heterogeneity of people causing the differentiated norm engagement, we introducing the following characteristics for the guest agents: being a regular or non-regular guest at a restaurant, having a sensitive taste, and having a high or low social drive. Table 4.3 explains these in more detail.

The distinction between regular guests and non-regular guests is relevant, because different types of guests feel differently impacted by the norm. Regular guests think that they have a spot guaranteed, and thus do not perceive a threat by the norm, compared to guests (non-regular guests) that do not think that they have a spot guaranteed. They perceive a thread by the norm.

The distinction between people that are affected by the change of the quality of ingredients of the restaurant owner (sensitive taste) is important, because this is a reaction to the behavior of the restaurant owner. The actions of the agents do not

Table 4.2: Needs of the guest agents with their range and their threshold (minimum level of satisfaction, recall Section 3.3.1, determined through model calibration). For belonging, pleasure, and leisure, the high social drive threshold is mentioned first (first line in the respective cell) and the low social drive later (second line in the respective cell). The thresholds for those needs follow a Gaussian distribution in that specified range

Need	Description	Range	Threshold
Belonging	Need to go out and meet with friends	$[0 \ldots 1]$	$0.7 \leq threshold \leq 0.9$ $0.4 \leq threshold \leq 0.6$
Pleasure	Need to eat nice food outside	$[0 \ldots 1]$	$0.7 \leq threshold \leq 0.9$ $0.4 \leq threshold \leq 0.6$
Leisure	Need to relax at home	$[0 \ldots 1]$	$0.4 \leq threshold \leq 0.6$ $0.8 \leq threshold \leq 1$
FinancialStability	Need to have enough money for living	$[0 \ldots 1]$	1
Compliance	Need to fulfill contractual obligation to go to work	$[0 \ldots 1]$	1
Sleep	Need to sleep in the night	$[0 \ldots 1]$	0.953
Security	Need to feel secure to have a place at the restaurant when going out	$[0 \ldots 1]$	0.953

.

Table 4.3: Overview of differentiating characteristics of the guest agents.

Characteristic	Description	Effect in the model
Regular guests vs. Non-Regular guests	Regular guests are guests valued (due to coming to the restaurant constantly and frequently to the restaurant) by the restaurant owner, as opposed to non-regular guests	Regular guests expect that they have always their spot guaranteed, and thus their security need satisfaction does not drop when the norm becomes active, as they do not feel impacted by the norm, whereas non-regular guest feel negatively impacted by the norm as their spot is not guaranteed
Sensitive taste	When the restaurant owner is lowers the quality of ingredients some guests experience that as a negative change (less tasty food), while others do not	The guests with a sensitive taste get a lower satisfaction of their pleasure when going to the restaurant if the quality of ingredients is changed, while the need satisfaction stays the same for the other guest agents that have no sensitive test.
High social dining tendency vs Low social dining tendency	This distinction is made to have guests needing go more often (e.g. twice a week) and some go out less often (e.g. once per week)	High socials have a higher threshold for their belonging and pleasure needs which also deplete faster, their leisure need has a lower threshold and depletes slower. This is vice-versa for the low socials.

happen in a vacuum. They are also affecting other agents. When the restaurant owner is lowering the quality of ingredients, some guests might get less pleasure form going to the restaurant and eating at it, while others do not feel a change (see here the restaurant owner specification in the next section).

Finally, the distinction between people that go less out to the restaurant to eat (low socials) and people that go out more often to the restaurant (high socials), is relevant, because this is another way of investigating agents that might be affected differently by the norm.

Having these characteristics allows us to analyze the results of the scenario experiments from various viewpoints. Each member of each characteristic can be combined with all members of the other characteristics resulting in $2 \times 2 \times 2 = 8$ distinct groups of guest agents in total:

1. $\{regular\ guest \wedge have\ sensitive\ taste \wedge low\ social\}$
2. $\{regular\ guest \wedge not\ have\ sensitive\ taste\ \wedge low\ social\}$
3. $\{regular\ guest \wedge have\ sensitive\ taste \wedge high\ social\}$
4. $\{regular\ guest \wedge not\ have\ sensitive\ taste \wedge high\ social\}$
5. $\{non - regular\ guest \wedge have\ sensitive\ taste \wedge low\ social\}$
6. $\{non - regular\ guest \wedge not\ have\ sensitive\ taste \wedge low\ social\}$
7. $\{non - regular\ guest \wedge have\ sensitive\ taste \wedge high\ social\}$
8. $\{non - regular\ guest \wedge not\ have\ sensitive\ taste\ wedgehigh\ social\}$

All of these characteristics of agents share the same actions, see Table 4.4. The table shows that the reserving a spot in the restaurant action is only available in the final scenario where the agents can circumvent or violate the norm. This is an abstraction we made for the model to show clearly a potential reaction to the size-based restriction norm. We acknowledge that in reality some people might also reserve a spot even though a norm like the size-based restriction is not active. However, this is not relevant for the purpose of this model and the scenarios. The key point that we want to show is that people that are negatively impacted by the norm might change their behavior. One such action can be to reserve a spot in the restaurant, if previously not done. We could have also used another action, such as changing one's eating times. But the general outcome would be the same, namely showing that people might adapt their behavior as a response to the new norm. It is also fair to assume in this case, that the people who reserved a spot before the size-based restriction norm, would continue doing so after the norm became active. Therefore, they can be seen as having a spot guaranteed, and thus would fall in the impact category of the regular guests, as they would not feel impacted by the norm. Consequently, we see this abstraction as fair to make, as the purpose is to observe an increase in the number of reserved spots, compared to the scenario where the norm is not active, or the norm is active but people cannot do anything against the perceived negative impact of the norm. Because of this, we do not see this abstraction being a thread to the validity of the model, but rather a fair design choice to make.

Table 4.4: The actions available for the guest agents. The yellow background indicates that the action is only available in the Norm response diversity scenario (see Section 4.4) while the other actions are available in every scenario.

Action	Description	Affected needs
Work	The agent works at their working place. This action can only be chosen during working time.	+ Compliance + Financial Stability - Leisure - Sleep
Stay at home	The agent is at home on the couch (reading or watching TV, or similar). This action can be chosen at any time during the day but only when the agent is at home or work (including then going home from work).	+ Leisure
Sleep	The agent sleeps in their bed. This action can only be chosen during the night.	+ Sleep
Go to the restaurant *without* reservation	The agent goes to their restaurant and tries to see if there is a spot available for them to eat. This action can only be chosen during the early evening and when they are not at a restaurant or park currently.	+ Belonging + Pleasure - Financial Stability - Leisure - Sleep
Go to the restaurant *with* reservation	The agent goes to their but this time they try to reserve a spot before they go, such that they have a spot guaranteed when they go. This action can only be chosen during the early evening, and when the agent is not currently at a restaurant or park.	+ Belonging + Pleasure + Security -Financial Stability - Leisure - Sleep
Go home from the restaurant	The agent goes home from the restaurant, this includes paying for the restaurant visit. This action can only be chosen if the agent is currently at a restaurant.	+ Belonging + Pleasure - Financial Stability - Leisure - Sleep

Continued on next page

Table 4.4 – continued from previous page

Action	Description	Affected needs
Go to the park	The agent goes to the park. This action can only be chosen in the afternoon or the early evening and when the agent is not currently at a park or restaurant.	+ Belonging + Pleasure + Security - Financial Stability - Leisure - Sleep
Go home from the park	The agent is goes home from the park. This action can only be chosen when the agent is currently at a park	+ Belonging + Pleasure - Financial Stability - Leisure - Sleep

The physical times (physT) and social times (socT) for the guest agents during the week and the weekend can be found in Table 4.5.

Table 4.5: Physical time (physT) and social time (socT) for the guest agents for during the week and the weekend.

Physical Time	**Social Time Week**
Morning	Working Time
Afternoon	Working Time
Early Evening	Free Time
Late Evening	Free Time
Night	Sleeping Time

(a) Time schedule of the guest agents during the week, Monday to Friday

Physical Time	**Social Time Week**
Morning	Free Time
Afternoon	Free Time
Early Evening	Free Time
Late Evening	Free Time
Night	Sleeping Time

(b) Time schedule of the guest agents at the weekend, Saturday and Sunday

To enhance the model's realism, we introduced a restaurant-switching mechanic inspired by real-life behavior. Instead of going home when denied a spot, guests may attempt to dine at another restaurant, depending on their social drive. High social drive guests (high social guest agent characteristic)) denied a spot will try another restaurant in the same time step, while low social drive guests will go home and try again the next time.

Initially, guests keep their original restaurant for the next time, but if high social drive guests get their spot denied twice, they permanently switch to the new restaurant. Low social drive guests require four denials to switch. Regular guests immediately switch after one denial, feeling 'hurt' by the rejection, but they become non-regular guests at the new restaurant, as we haven't defined a way for guests to regain regular status.

The switching process follows a sequential order: guests from restaurant one try restaurant two, then three, those from restaurant two try restaurant three, then restaurant one, and guests from restaurant three try restaurant one and then two. This mechanic is designed to increase realism and address concerns of abstraction in the model but is not part of the agent deliberation architecture, and therefore, just model specific.

4.3.1.2 Restaurant Owner Agent Make-Up

In this section we specify the restaurant owners including their needs, priorities in values, available actions, characteristics to introduce heterogeneity, and their social time and location database. The restaurant owner agents have the following needs Table 4.6. Similar to the guest needs, the restaurant owner needs are also inspired by our ASSOCC project [75] which is based on Maslow's hierarchy of needs [94], and then adopted for this specific model.

Table 4.6: Restaurant owner needs with their description, range, and thresholds (minimum level of satisfaction, recall Section 3.3.1), determined through model calibration). The thresholds for the different restaurant owners are mentioned in the order of restaurant owner 1 / restaurant owner two / restaurant owner three

Need	**Description**	**Range**	**Threshold**
Financial Stability	Make a financial living with the restaurant	$[0 \ldots 1]$	0.4 / 1 / 1
Security	Having good customer relations with one's regular guests	$[0 \ldots 1]$	1 / 0.5 / 1
Compliance	Adhere to the norm (always satisfied when the norm is not active)	$[0 \ldots 1]$	1
Self-Realization	Having high quality dishes (focus on ingredients of the dishes)	$[0 \ldots 1]$	1 / 0.5 / 0.5

There are three restaurants and consequently three restaurant owners in the simulation. We make each restaurant owner distinctive from one another as follows:

Restaurant owner one: Restaurant owner one reflects someone that opened the restaurant out of hobby to provide good quality experience to the guests (high priority in self-realization and lower priority in finance). This also means that their regular guests are very important for them, as they really care about their guests, and value if they return on a regular basis. This results in a high threshold for the security and self-realization needs, and lower threshold for the financial stability need. The priorities in Schwartz values [138] for restaurant owner one are as follows: $[Security, Conformity, Tradition, Benevolence, Universalism, Power, Achievement, Hedonism, Stimulation, Self-Direction]$. In this way it is assured that the restaurant owner is prioritizing actions that are caring for their regular guests and also prioritizes running the restaurant in the way they want to (using high quality ingredients), without focusing on the money. Furthermore, restaurant owner one is owning the place of their restaurant meaning that they do not have to pay rent for their restaurant and thus have lower costs.

Restaurant owner two: Restaurant owner two opened the restaurant purely based on monetary motivation (the guests are just wallets) and the food quality is not that important. This results in high thresholds for the financial stability need and also compliance need (as the restaurant owner does not want to pay the fine), and a lower threshold in the security need (regular guests are not important to the restaurant owner, money first regardless from whom) and self-realization (as the restaurant owner prioritizes money over the quality of the dishes). The priorities in Schwartz values [138] for restaurant owner one are as follows: $[Power, Achievement, Security, Conformity, Tradition, Hedonism, Stimulation, Self-Direction, Benevolence, Universalism]$. In this way it is assured that the restaurant owner is prioritizing money over customer relations while also obeying to the norm. This reflects the money-focused restaurant owner. Furthermore, restaurant owner two is renting the place of their restaurant meaning that they have to pay rent for their restaurant and thus have higher costs.

Restaurant owner three: Restaurant owner three is a mix of restaurant owner one and restaurant owner two, having a high priority on both, good customer relations with their regular guests, and making as much money as possible. This results in a high thresholds for the needs of financial stability and security, and a lower threshold for the need of self-realization. The priorities in Schwartz values [138] for restaurant owner one are: $[Security, Power, Achievement, Conformity, Tradition, Hedonism, Stimulation, Benevolence, Universalism, Self-Direction]$. This order ensures that the restaurant owner priorities money, as well as their connection to that regular guests, as those values have the highest priorities. Furthermore, restaurant owner three is renting the place of their restaurant meaning that they have to pay rent for their restaurant and thus have higher costs.

A restaurant owner is evaluating and updating their financial stability need every week. The reason for this is that restaurants can have days with fewer guests and days with more guests. Since this these days and fluctuations are usually known to the restaurant owner, they will not make their decision to change something based on a day with fewer guests, but rather look at their financial in set intervals, in the model every week on a Monday morning, to see if the restaurant is making money or losing money. This is why the need satisfaction looks like a step-wise function in the graphs, based on the fact if the restaurant owner making money (step upwards) or losing money (step downwards).

In addition, we also made a difference in the depletion of the financial stability need to account for varying importance of that need (similar to the depletion of the pleasure and belong need for high social and low social guest agents). Money is not that important for restaurant owner one, and therefore they perceive making money in a week and losing money in a week as having the same severity. Thus their need satisfaction is increased by 0.1 if they make money, and reduce by 0.1 if they lose money. For restaurant owners two and three money is very important, and therefore losing money is perceived worse than making money. This means that if they lose money in a week they lose 0.2 of their financial stability need satisfaction, while they gain 0.1 for their financial need satisfaction for making money in a week. Making money in a week means to have more money on a Monday morning than on the Monday morning of the previous week. Losing money in a week means to have less money on a Monday morning than on the Monday morning of the previous week.

The self-realization is updated every time an action that is changing the quality of ingredients is performed by the restaurant owner agent. Otherwise, the need stays constant at the level determined by the action. The security need is updated once (drop) after the size-based restriction norm becomes active, and then gets updated if an action is performed to let in guests more than allowed. The compliance need is always satisfied and only drops if the restaurant owner agent is getting caught violating the norm. Then it will be satisfied again when the agent pays the fine, which in our model is done immediately.

We acknowledge that using cheaper ingredients should also negatively impact the security need of the restaurant owners, as they can be less sure if their guests keep coming to their restaurant. But this would require more detailed modeling of the restaurant owners in terms of their knowledge. Examples include knowledge about how many of their guests have a sensitive taste, how often those guests with a sensitive taste come to their restaurant, and subsequently how much money the restaurant owners would lose if the guests with sensitive taste would avoid the restaurant after finding out about the use of cheaper ingredients. But this is out of the scope for this model, as it is not necessary for the purpose of our model to verify and validate our proposed PBADA agent deliberation architecture, as with this model we can already that our architecture is indeed able to accommodate for situated and differentiated norm engagement (perspectives on norms) thereby allowing for various norm responses (obedience, obedience and circumventions, and violation).

Since we already touched upon the actions in relation to their needs, we want to describe them in more detail now. A summary of the actions can be found in Table 4.7. The restaurant owners have a range of actions that they can chose from to respond to the size-based restriction norm. One possibility is to continue the restaurant the way they used to and change nothing, reflected in the business as usual action. If the restaurant owners want to do something against the loss of income but still want to comply (obey) to the norm, they can chose to use cheaper ingredients for their dishes, as it is not forbidden by the size-based restriction norm. This increase the profit they make per dish, but also reduces the pleasure that the guests get from eating at the restaurant (if they have a sensitive taste) and the self-realization need of the restaurant owner. The restaurant owner agents can also violate the norm, and let in more guests than allowed, according to the size-based restriction norm. This can either be only letting regular guests in more than allowed or letting everyone in more than allowed. With this, the restaurant owners do not have to reduce the quality of their ingredients, and can ensure good customer relations, especially with their regular guests which fulfills their security need. However, this also means that when they get caught violating the norm, they have to pay a fine, and their compliance need drops.

Table 4.7: The actions available for the restaurant owner. The yellow background indicates that the action is only available in the Norm response diversity scenario while the other actions are available in every scenario.

Action	Description	Affected needs
Business as usual	The agent is doing their daily business at the restaurant (managing, attending to the guests, etc.) without any changes of the current status quo. This action can be chose at any time during the day.	
Let it no more guests than allowed	When the space limit is reached (according to the norm), no more guests are allowed in the restaurant. This action can be chose when the restaurant owner previously decided to let in regular guests or everyone in more than allowed.	+ Compliance - Security
Let only regular guests in more than allowed	The agent is letting in regular guests (but not non-regular guests) even though the restaurant might be full according to the space limit of the norm. This action can be chosen when the restaurant owner previously did not let in anyone more than allowed or everyone in more than allowed.	+ Financial stability + Security - Compliance
Let in everyone more than allowed	The agent is letting in everyone (regular guests and non-regular guests) even though the restaurant might be full according to the space limit of the norm. This action can only be chose if the restaurant owner previously decided to not let in anyone more than allowed or only regular guests more in than allowed.	+ Financial stability + Security - Compliance
Use high quality ingredients	The agent changes to using the highest quality of ingredients for their food at their restaurant (the default at the start of the simulation). This action can only be chosen if the restaurant owner previously used cheaper ingredients.	+ Self-Realization - Financial Stability

Continued on next page

Table 4.7 – continued from previous page

Action	Description	Affected needs
Use cheaper ingredients	The agent changes to using cheaper ingredients for their food at their restaurant (lower quality of ingredients to increase their profit). This action can only be chose if the restaurant owner previously used high quality ingredients.	+ Financial Stability - Self-Realization

4.3.2 Model initialization

In this section, we describe the initialization of the model. This includes the notion of time, the specification of the norm, as well as the amount of guests and restaurant owners, and how they are distributed.

A timestep in the simulation represents a distinctive point in time in the simulation. For our model, we chose the following meaning for a timestep: morning, afternoon, early evening, late evening and night. These five timesteps then make up one day, and seven days make up a week. This is to go with the analogy of the real world. Finally, we decided to let the simulation run for six months which resulted in 910 timesteps in total. This is to show the short-term and potential long-term effects that the norm can have rather than having the simulation run only for a short period of time. The locations existing in the simulation are the home of the guest agents and the restaurants where the guests can go and the restaurant owners can live and work. Furthermore, we do not model direct movement as it is not relevant for the purpose of our model, and thus, the guests teleport from one place to another place between the timesteps. For the agents, we have 600 guests agents, with 200 assigned to each restaurant, see Table 4.8. We also assign the guests to specific parks. This is to better show the effects of people interacting with the norm, and reacting to the behavior of other agents.

Given that there are 600 guests and 200 guests per restaurant, there are a total of three restaurants in the simulation. This is in line with the three different restaurant owners described above. The reason that we have three restaurants is to show a diversity of norm responses, such as violating the norm and circumventing the norm or a combination of both. For each restaurant there is also a park, meaning that there are three parks in the simulation. With the assignments the guests assigned to restaurant one can chose between going to restaurant one and park one, guests assigned to restaurant two can chose between going to restaurant two and park two, and guests assigned to restaurant three can chose between going to restaurant three and park three. The reason for the different sizes and different numbers of regular guests per restaurant is to further differentiate the restaurant owners from one another by also varying their conditions (available spots, and number of regular guests).

The norm itself is becoming active after nine weeks in the simulation (at timestep 315) and stays active for the remainder of the simulation. The size-based restriction limits the available seats per restaurant to a capacity of 50%. The fine for violating the norm is 5000 money units. With our proposed ADICDIRO framework (recall

Table 4.8: Distribution of guest agents per restaurant

Restaurant owner	**Number of seats**	**Number of assigned guests**	**Number of regular guests**
1	126	200	80
2	76	200	15
3	100	200	30

Section 3.4), we can describe the size-based restriction norm, $n_1 \in \mathcal{NO}$ (with $\mathcal{NO}$ = {size-based restriction}), as follows:
Attribute (A^{n_1}) = Restaurant owner
Deontic (D^{n_1}) = must not
$I^{n_1}_{Verb}$ = let more than 50% of their guests in
$I^{n_1}_{Object}$ = their restaurant
Deontic + aIm ($DeoaIm^{no_1}$) = $\{\#guests \leq \frac{capacityoftherestaurnt}{2}$, $\#guests > \frac{capacityoftherestaurnt}{2}\}$
Condition (C^{no_1}) = {timestep == 315 , none}
Deadline (Dl^{no_1}) = time==0
Repair (R^{no_1}) = Make guests leave
Punishment (O^{no_1}) = Pay 5000 money units fine
promValues ($promValues^{no_1}$) = Safety
demValues ($promValues^{no_1}$) = Power, Achievement, Security
affectedNeeds ($affectedNeeds^{no_1}$): Security, FinancialStability, Belonging

The reason for letting the norm active for such a long time (about two thirds of the whole simulation run in our case) is to show that there are no long term consequences that differ from the short-term effects of the norm.

4.3.3 Process Overview

Figure 4.1 provides a process overview of what is going on at each timestep of the simulation. Since the execution order for each deliberation step of the agents needs to be sequentially, the restaurant owners are deliberating first before the guest agents do. Because of this, the decision of the restaurant owners to let more people in than allowed is meaning that the respective agents are not counting towards the seat limit if the norm is active. For example if the restaurant owner is willing to let regular guests in more than allowed, it means that they are not counted towards the seating limit, and thus, only non-regular guests would be considered when the guests check if there is still space in the restaurant. But there is no direct communication between restaurant owners and guests when the guests try to go to the restaurant. Of course, when checking for potential norm violations, all guests are counted.

Furthermore, the two elements highlighted in color, the update of the financial stability need of the restaurant owner and the security need update of the restaurant owner, do not happen at every timestep while the other events do happen at every timestep. The financial stability need update happens every Monday morning, i.e. once a week, for the restaurant owner to see if they made or lost money in the last week. The security update of the restaurant owner happens only when the size-based restriction norm becomes active, as the satisfaction drop of that need is directly connected to the norm being active.

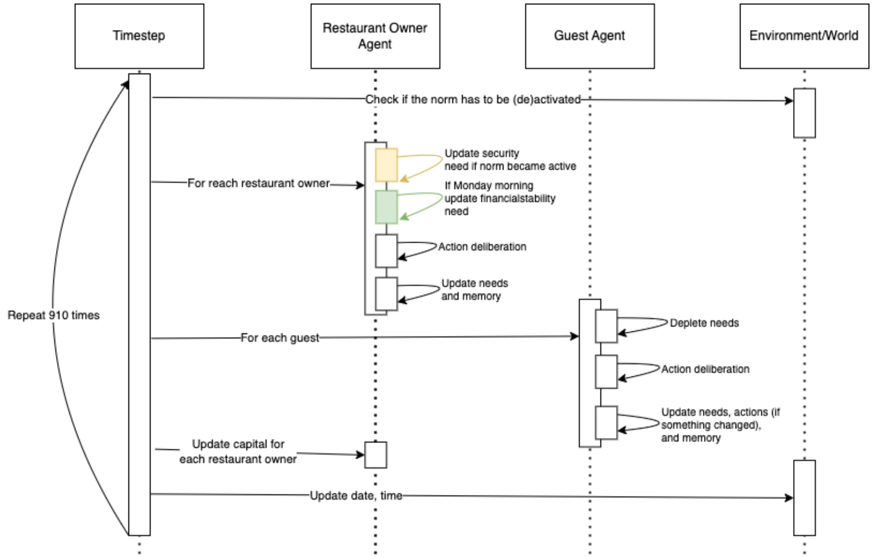


Figure 4.1: Process overview of what is going on during each timestep in the simulation. The update of the security need and the financial stability need of the restaurant owner are highlighted, as this is not happening at every timestep. The other events are happening at the every timestep.

4.3.4 Implementation Platform

The implementation of the model is done in Unity[1]. We chose Unity over other existing agent-based social simulation software, such as NetLogo [162] or Repast [117], due to the visual demands we have for supporting policymakers. Strong visual support, and advanced interaction possibilities allow users to modify the simulation and get the insights they need. Something that Unity, as a game engine, is providing by design.

4.4 Experiment Design - Scenarios

To systematically verify and validate our proposed PBADA agent deliberation architecture, and engage in systematic experimentation (based on Kleijnen [86]), we need different scenarios. Each scenario alters one factor while the others stay fixed. We

[1]https://unity.com/

differentiate between two factors: the size-based restriction norm (being active or deactive), and the actions available to the agents. This can be translated into the following three scenarios:

Scenario 1: Norm not active (baseline)

Scenario 2: Norm active but available actions the same as in the previous scenario (resulting in the norm being a limitation on the agent's behavior)

Scenario 3: Norm active and more actions available to the agents (norm response diversity)

For each scenario, expectations were formulated[1]. Regarding the results themselves, we split each expectation for each scenario into required patterns, and analyzed the results to see if each pattern was met. When every required pattern for the expectation in a specific scenario was met, we considered the corresponding expectation to be met. During this analysis we used the trace validity suggested by van den Hurk et al. [74].

In general, our expectations are oriented at patterns that one would expect in real life. This is to show that our agent deliberation architecture can produce *plausible* behavior. Plausibility is very important, because one purpose of modeling is to highlight the underlying, in our case normative, decision-making mechanisms of the agents [42]. Furthermore, since we are aiming to show plausible results for our V&V process, we call the desired behavior expectations[2] rather than hypotheses, because we want to show plausible and intuitive behavior that can be expected in reality rather than hypothesizing about potential behavior. Furthermore, we start from the expected behavior in reality rather than exploring an unknown phenomenon. A summary of the different scenario can be found in Table 4.9.

The **Norm not active (baseline)** reflects the life as normal case where the norm is not active and the people go about their daily lives. The purpose of this scenario is to show that our agent architecture is able to show such 'normal' behavior. Furthermore, this scenario serves as a baseline to compare the results and behavior of the agents in the other scenarios against. Having such a baseline is important to draw any conclusions from the other scenarios. To check the behavior of the agents for plausibility, we formulate the following expectation for this scenario:

Expectation 1 (Baseline Scenario)
The needs of the guests and restaurant owners are well satisfied. The guests go to the restaurant between 0.5 (once every two weeks) to twice per week. Only a few spots are denied (≤ 10%). The restaurants are profitable, and the restaurant owners can run the restaurant the way they want to.

Note that we are not interested in the exact numbers, but rather the overall pattern of behavior, and thus we refrain from providing exact numbers. In many ways meeting

[1]These are for the validation. For the verification they can be found in verification section (Section 4.5. They are quite similar however.

[2]Note that these expectations are also falsifiable

the expectations for this scenario can be seen as a calibration exercise. Note here that seeing this scenario as a calibration exercise does not take away from its contribution to the validation of our agent deliberation architecture. In Lorig et al. [91], we also pointed out that calibration is a form of validation. Furthermore, we used such a baseline in our ASSOCC project [37] to calibrate our model by letting the simulation run without infections.

The **Norm active but available actions the same** scenario is the scenario where the norm (size-based restriction) is introduced, The purpose of this scenario is to show that the agents using our agent deliberation architecture act on the size-based restriction norm. This means that the agents decide on their action based on their underlying characteristics. But crucially, we keep the available actions the same to be able to isolate the behavior of the agents as best as possible to be able to ground every differing behavior compared to scenario one (baseline scenario) in the norm being (de)active. Furthermore, this scenario highlights the challenges of the limiting take on norms: a norm is only a restriction on the behavior of the agents. Our expectation for this scenario are as follows:

Expectation 2 (Norm as limitation)
The guests that are not affected by the norm (regular guests) do not alter their behavior. The guests that are affected by the norm (non-regular guests) alter their behavior as another already existing alternative action (going to the park) is more attractive now. More guests get their spots denied compared to the previous scenario. The restaurants go broke, and do not recover form that, as the restaurant owners can only abide the norm.

The **Norm active and more actions available to the agents** scenario shows the case where the norm is active and the agents have more actions available to them. The purpose of this scenario is to show the full power of our proposed PBADA agent deliberation architecture. This means adding the realism of different norm reactions. In the previous scenario the agents could only obey the norm, and in this scenario the agents are also now able to violate the norm, and circumvent the perceived negative consequences of the norm, i.e. obeying the norm but also taking actions against it at the same time. Compared to the previous scenarios, adding more actions to the agents allows us to see that the agent architecture works and the agents decide to take actions against based on their motivations and specifically as a reaction to the size-based restriction norm. We can make this comparison, because in the previous scenario we ensured that the agents are taking the norm into account, and thus this factor of the additional actions is now as isolated as possible. Our expectation for this scenario are as follows.

Expectation 3 (Norm response diversity)
The restaurant owners will violate the norm by letting in more guests than allowed and/or use cheaper ingredients to increase their profit. The restaurants are profitable. The guests not affected by the norm (regular guests) do not alter their behavior. The guests affected by the norm (non-regular guests) reserve a spot at the restaurant. The guests affected by the ingredient change (having a sensitive taste) react to it and go to the park.

Table 4.9: Mapping of the norm reactions to the scenarios

Scenario	Name	Description
Scenario 1	baseline	No size-based restriction norm active, the restaurants can operate on their full capacity. Actions for the guests include for example going to the restaurant *without* reservation, and going to the park (see Table 4.4), and for the restaurant owner doing business as usual (see Table 4.7)
Scenario 2	Norm as limitation	The size-based restriction norm is active after 9 weeks (315 timesteps) limiting the restaurant capacity to 50%. The agents can only obey to the norm. They have the same actions available as in scenario 1. They cannot take any actions against the perceived negative consequences of the norm.
Scenario 3	Norm response diversity	The size-based restriction norm is active after 9 weeks (315 timesteps) limiting the restaurant capacity to 50%. The agents actions include violating the norm or take other (norm obeying) actions to counteract the perceived negative consequences of the norm, in addition to the actions in scenario 2. Additional actions are for the guests going to the restaurant *with* reservation (see Table 4.4), and for the restaurant owner to letting for example regular guests in more than allowed or using cheaper ingredients (see Table 4.7)

Each scenario is repeated 20 times with a different random seed for determining the initial need satisfaction of the pleasure, belonging, and leisure needs of the guest agents, as well as the guest agents assigned to having a high or low social drive. The assignments of the regular and non-regular guest characteristic, as well as being potentially affected by the ingredient change (i.e. having a sensitive taste) were kept constant, as the key element in our proposed PBADA agent deliberation architecture are the needs, and thus the focus and variations via the different random seeds is on them. The scenario is repeated 20 times, because this resulted in a stable behavior and further runs did not alter the results anymore.

4.5 Verification of the Agent Deliberation Architecture

Verification means that the implementation is reflecting the underlying architecture correctly. This is reflected in Gilbert & Troitzsch [55] saying that the model works according to our (the modeler's) expectations. To do this, we take a pragmatic approach by forming expectations for the behavior of the agents in each scenario and contrast the behavior of the agents between the different scenarios. This is to single out the behavior of representative agents for each perspective, and analyze their behavior to see if it matches our expectations given our PBADA architecture.

4.5.1 Guest agents

4.5.1.1 Scenario 1

For the guest agents, we chose a non-regular guest with a high social drive, and no sensitive taste to single out their behavior. This guest agent is representative for all guest agents, as the norm is not active, and the restaurant owner has no further actions than business as usual available. For this guest agent, we expected them go out twice per week, once earlier in the week, and once later in the week or at the weekend. Furthermore, we expected them to go to work in the morning and afternoon (Monday to Friday), as it is working time, and the most salient action to do during that time is to go to work.

Concretely for the architecture, this means that the base level, the most complex deliberation level for encountering a novel situation, and the habitual level for being in a known situation are used, see Figure 3.5.

When looking at the behavior of the particular guest agent, we can see that in a novel situation on Monday morning (i.e. the first step of the simulation in the first week), the agent is going to work. This is expected behavior, and shows that the path of a complete novel situation in the architecture works. Furthermore, on Monday evening this guest agent is going to the restaurant rather than staying home, as it is their most salient action. Then on Thursday evening the guest goes to the restaurant for the second time per week, as their belonging and pleasure needs are depleted again (given their high social drive), and thus they go out again. Following the agent, the behavior repeats every week, showing that the habitual deliberation part is working as well. Finally, different needs are salient for the agent at different timesteps showing that the base level is working as well.

4.5.1.2 Scenario 2

For scenario two, the size-based restriction norm is active now. We kept the same guest agent, but also required a second one. The first guest agent from the previous scenario is a non-regular guest, and thus affected by the norm. To make sure that this part works correctly, we also need a guest agent that is a regular guest, and thus, not affected by the norm.

We expect that a guest agent that is affected by the norm, i.e. a non-regular guest, to change their behavior and go to the park instead of the restaurant, as it provides security for getting a spot. For a guest agent that is not affected by the norm, we expect that they do not change their behavior and keep going to the restaurant. In terms of our PBADA architecture this means that the path of deciding on the potential negative impact of the norm is used, see the upper path of the complex deliberation level in Figure 3.5.

We can observe that the non-regular guest agent is altering their behavior after the norm becomes active. When the time comes for the agent when they would normally go to the restaurant, they move into the complex deliberation where they deliberate about the negative consequences of the norm. Their they decide they go to the park, and thus chose the expected action, as the park provides a security need satisfaction gain, with the security need being the one negatively impacted by the norm. This is a different behavior compared ton the previous scenario.

To make sure that this behavior comes out of the deliberation of the agent, and is not an artifact, we look at a random regular guest agent, and see that this agent is not changing their behavior and keeps going to the restaurant.

4.5.1.3 Scenario 3

For the third scenario, the agent has now the action available to reserve a spot in the restaurant. The final characteristic that is relevant here is that the agent does not have a sensitive taste, and is therefore not affected by the change of the quality of ingredients by the restaurant owner. This is relevant, as we expect agents to go to the park if they are affected by the ingredient change of the restaurant owner.

For the guest agent that we looked so far at, we expect that they will go to the restaurant and reserve their spot there. This is a change to the previous scenario, as they now have the action to reserve a spot there which provides the security need satisfaction gain and overall the higher need satisfaction gain, as the guests get more pleasure from going to the restaurant than going to the park. This is indeed what is happening in the simulation. Instead of going to the park the guest agent is going to the restaurant again, and selects the action to reserve their spot.

We also have to compare this behavior to another random regular guest that has a sensitive taste, and is therefore affected by the ingredient change. We expect, and see that they go to the park, even though they could reserve a spot in the restaurant. This behavior is expected is and important. Confirming this behavior makes sure that the medium complex deliberation path of finding an alternative action (see Figure 3.5) is working as expected. We ca confirm this, because the guest is a regular guest, and thus not affected by the norm. This means that they would not change their behavior. But since they have a sensitive taste, they are affected by the action of the restaurant owner, and thus find an alternative action.

4.5.2 Restaurant owner

4.5.2.1 Scenario 1

For the restaurant owner agents, we will look at restaurant owner one in more detail. This agent is representative, as the norm is not active, and only one action is available for the restaurant owners. The restaurant owner has only the action available of business as usual. This is the only one they can chose, and consequently do so. We expect for this scenario that the needs of the restaurant owner are well satisfied, as they can have all the seats available for their guests, make money with their restaurant, and use high quality ingredients.

This is the case, as all the needs are at the maximum satisfaction level of one. The agent deliberation architecture parts are the same as for the guest agent in this scenario.

4.5.2.2 Scenario 2

For scenario two, the restaurant owner has still only the action of business as usual available. Since the size-based restriction norm is active now, we expect that their need satisfaction of their security and financial stability needs drop, as they go broke and cannot take care of their customer relations.

This is indeed what is happening in the simulation, as restaurant owner one, can only select the business as usual action which they do. The agent deliberation architecture parts are the same as for the guest agent in this scenario.

4.5.2.3 Scenario 3

For the third and final scenario the restaurant owner can now let in more guests than allowed and use cheaper ingredients. Given that restaurant owner one is strongly customer focused on their regular guests, and money is only the second priority, we expected in this scenario that restaurant one will first try to satisfy the security need (relevant for customer relations) regardless of violating the norm or not, and then their financial stability need.

This is what is happening in the simulation. After the size-based restriction norm becomes active, the satisfaction of the security and financial stability needs of the restaurant owner drop. For restaurant owner one, the security need is now urgent to satisfy while the financial stability need is still satisfied. Restaurant owner satisfies the security need by allowing regular guests in more than allowed. Since this also generate enough income for the restaurant owner, as they do not lose money anymore, they do not take any further action. This is expected behavior, and use the complex deliberation level of dealing the perceived negative consequences of the norm, see Figure 3.5.

We can contrast the behavior of restaurant owner one to the behavior of restaurant owner two. This is important, as restaurant owner two has a strong focus on money, and thus we expect them to chose a different action, namely using cheaper ingredients, as this satisfied their financial stability need. When observing the behavior of restaurant owner two, this is the case. They chose to use cheaper ingredients to stay profitable. Furthermore, they decide to not violate the norm, as the security need is still satisfied for them (even after the drop), as the need has a lower threshold for them.

The medium complex deliberation path of dealing the reactions to other agents is not relevant to show here, as it is not applicable to the restaurant owners in this model. However, it is very important to mention here that the medium complex deliberation path is already verified via the guest agent's behavior.

4.5.3 Conclusion

For each agent group, guests and restaurant owners, we looked at the expected behavior for each scenario and compared it between them. Since each of the behaviors of the agents is matching its corresponding expectations given the underlying PBADA architecture, we conclude that we successfully verified our PBADA agent deliberation architecture, as it is matching (the modeler's) expectations, and is therefore in line with the argumentation of Gilbert & Troitzsch [55].

Furthermore, we used representative agents for each group to make sure that every part of the our architecture works correctly. For the restaurant owner agent group, showing one restaurant owner was sufficient, as the behavior of the other restaurant owner agents works accordingly with their own need thresholds, as shown by looking at restaurant owner two. For the guest agents, we used a representative guest agent, and also compared the behavior to other guest agents with different characteristics if necessary, and showed that our PBADA architecture works as expected given then respective guest agent characteristics.

4.6 Validation of the Agent Deliberation Architecture

In this section, we will validate our proposed PBADA agent deliberation architecture, based on the systematic experimentation approach by Kleijnen [86] combined with the trace validity of van den Hurk et al. [74]. To do so, we will run each scenario from from Table 4.9, and then compare the results of the simulation runs of each scenario to the expectation that we have formulated for the respective scenario.

To do this, the expectations where formed in terms of trends that the results need to be showing. For example, Expectation 2 states that we expect the restaurants to go bankrupt. This needs to be seen in the results. But it does not matter for successfully meeting the expectation at which exact point in the simulation the restaurants go bankrupt, or if one restaurant goes bankrupt faster or slower than the another one.

To focus on these trends and to analyze the data, we used R [125] due to its widespread use, and functionality for statistical analysis [123]. The ggplot2 package [158] developed for R [125] is providing a smoothing function [123]. This smoothing function basically acts like a moving average. This enables us to look at the results, and see the trend at a glance [123]. Otherwise the results can be quite jagged which makes it difficult to get quick insights [123], and can also be distracting.

Lastly, we want to explain the usage of the words restaurant owner, restaurant, and park. To avoid cluttering the text, we use these words in the singular form when the distinction between e.g. each restaurant is not important. Instead of having to write the agents go to their respective restaurant, we simply write the agents go to the restaurant. When the distinction between the specific places and agents is important, we will use the exact restaurant number.

4.6.1 Scenario 1 - Norm not active

The first step of validation is a baseline of life as normal, so that later more complex scenarios can be pitted against this one. Therefore, we look at scenario one, where the norm is not active to establish a behavioral baseline. Recall, the expectations for scenario one

Expectation 1 (Baseline Scenario)
The needs of the guests and restaurant owners are well satisfied. The guests go to the restaurant between 0.5 (once every two weeks) to twice per week. Only a few spots are denied (≤ 10%). The restaurants are profitable, and the restaurant owners can run the restaurant the way they want to.

We will discuss meeting the expectations for each - the guests and the restaurant owner - separately, starting with the guests.

4.6.1.1 Guest Results

The parts of Expectation 1 that concern the guests involve their need satisfaction levels, their frequency of wanting to go and being able to go to a restaurant. To substantiate this the model needs to produce the following three patterns: 1) the needs of the guest agents are overall well satisfied (above their threshold). The satisfaction of all needs is relevant to show realistic human-like behavior, as we humans seek an overall balanced state, and as such want all our needs to be satisfied. 2) The people (guests) go to the restaurant in a realistic (between 0.5 to two times per week) manner [1], and 3) with rarely any spots being denied. We address now each of these parts separately.

1) Needs satisfaction: Figure 4.2 shows that all guest needs are highly satisfied with being above or around the 0.75 level and stable (no large fluctuations). To show that this means that the needs are well satisfied, we look at Figure 4.3 which shows the difference between each need satisfaction level and their respective threshold. In this figure, a positive value above zero means that the need is satisfied, with the current

[1]We acknowledge that not everyone is able to go to the restaurant every or every other week, but for the purpose of our model, we focus on the people that can

satisfaction level being above the threshold. If the value is negative, it means that the need is not satisfied, as the current satisfaction level is below the threshold. It is important to keep in mind that we look at smoothed averages over all agents here. This means for example in the case of financial stability that not all agents have this need not satisfied. Some have, and some do not, so on average the value can be a bit below zero. This can be the case when they go out in the evening and the need becoming salient again (as they spend their money when going out), and is then replenished the next day when they go to work in the morning and afternoon. Furthermore, the reason why the leisure need satisfaction is so much higher than the threshold is that it is the default action. Default action means that if all needs are satisfied at a given moment and the guest agents do not have a contractual obligation at this specific point in time (which can happen in our small example scale), the agents stay home (satisfies leisure need). This is a decision that we made by choice. However, it is not farm from reality, because when we do not know what to do, it is likely that we stay at home and do some leisure activity there.

While the results above meet the targeted pattern, the pleasure and belonging needs deserve a bit more attention as they differ slightly. Firstly, compared to the other needs in Figure 4.2, pleasure and belonging are lower. This is because of their lower threshold (due to the model calibration) Secondly, the pleasure need satisfaction is lower compared to the belonging need satisfaction, because some agents also go to the park during the day on the weekend. Going to the park provides less pleasure need satisfaction than belonging need satisfaction in general, and less pleasure need satisfaction compared to eating at the restaurant in particular, thus those needs diverge.

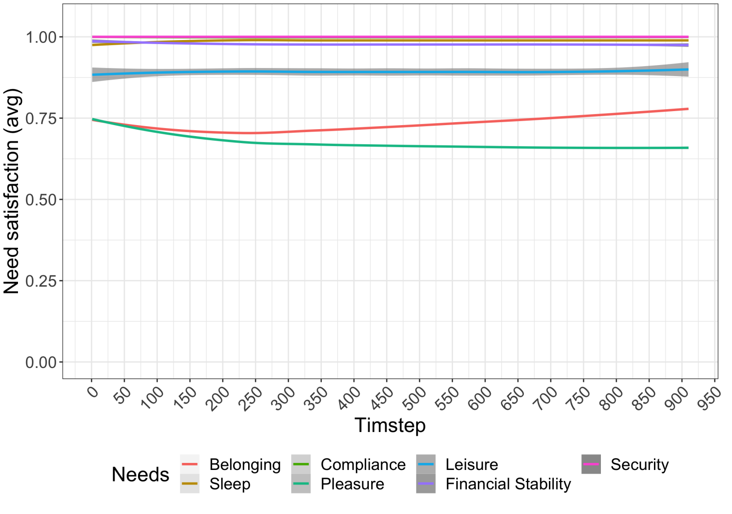


Figure 4.2: Average need satisfaction of the guest agents of all guests over time as average of 20 repetitions with smoothed function, baseline scenario

2) Going to the restaurant: The second pattern that needs to be met concerns the

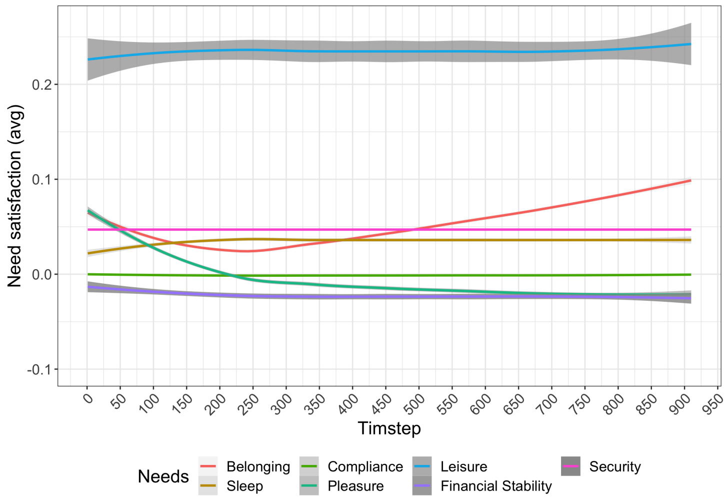


Figure 4.3: Average guest need to threshold satisfaction, value calculation: current fill level - threshold. Positive value: need is satisfied (above threshold). Negative value: need not satisfied (below threshold). Average of 20 repetitions with smoothed function, baseline scenario

frequency of guests visiting the restaurants (once every two weeks to twice per week). To address this part of the expectations, we look at Figures 4.4 and 4.5.

Figure 4.4 shows the number of people for each restaurant over time. We can see here that they are all spread out and display a regular pattern with the restaurants being completely full, as indicated by the black horizontal line showing the maximum number of spots available in the respective restaurant. Seeing a regular pattern here is important to show that when the size-based restriction norm is not active, the people are behaving in a regular pattern without any disruptions or sudden changes, as they would habitually do. The average trend lines are lower, as there are days with zero customers.

Figure 4.5 shows the average amount of restaurant visits for the agents of each restaurant per week. We can see here that for restaurant one, the agents go on average more than once per week, about 1.75 times with a few outliers below that. For restaurant two, not all agents go there at least once a week, with the box plot (green box) indicating that the agents go about 0.75 time per week. For restaurant three (blue box), the agents are going about one time per week to the restaurant. This behavior can be explained by the needs, and the action space available to the agents.

Given the varying importance of the pleasure and belonging needs (agent characteristic, recall Table 4.3), it can happen that the agents assigned to restaurant two have a lower priority of going out compared to the agents assigned to restaurant one. Furthermore, it can happen that the pleasure and belonging needs become salient on the weekend during the day, i.e. in the morning or afternoon. When this happens the

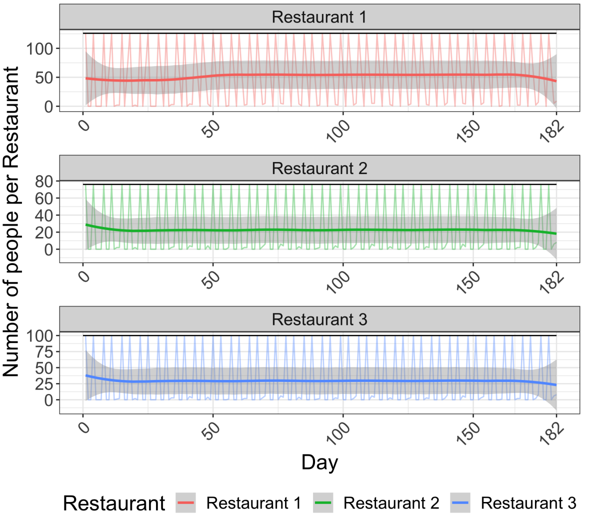


Figure 4.4: Count of all guest agents per restaurant over time. The black horizontal line presents the seat limit per restaurant. The trend line is low, as there are many days with zero customers which affect the calculation of that line. Average of 20 repetitions with smoothed function, baseline scenario

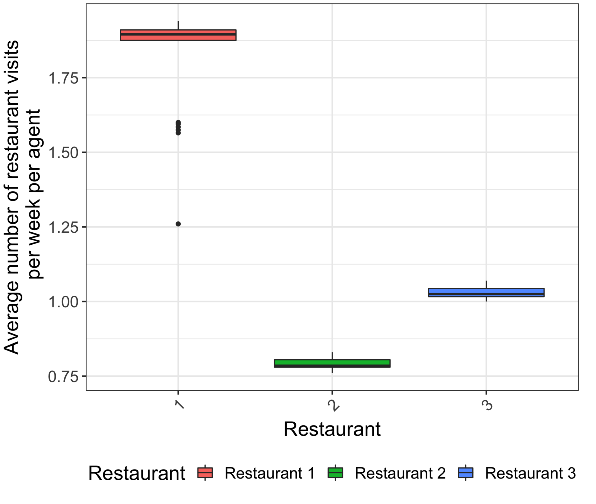


Figure 4.5: Average number of restaurant visits per agent per week for each restaurant, value calculation: total number of people per restaurant per week divided by the guest agents assigned to that restaurant, baseline scenario. Average of 20 repetitions, baseline scenario

agents can immediately go to the park to satisfy them, as they have no other commitments in the simulation. They can only go to the park in the morning and afternoon, because the restaurant is only open in the evening and not during the day. Together with the low social and pleasure drive, this means that for some agents it can be enough in some weeks to just go once to the park on the weekend during the day, and skip going to the restaurant for a week. It is also easy to see that if we would extend the model with commitments on the weekend for the agents, such as (non-food) shopping or other events, it would push the agents to go out in the evening to the restaurant, leading to the average number of restaurant visits per week rising to at least one.

Taking the information discussed above together, we conclude that the restaurant frequency expectation of people going to the restaurant once every two weeks to twice per week is met. The lower number of restaurant two of less than once per week does not pose a thread, as it is still more than once every two weeks (specifically 0.75 times), and can be explained by the characteristics of the agents, and the small action space available to the agents.

3) Spots denied: The final pattern for this expectation that we have to address is the amount of people that get their spots denied per week which we indicated as a few ($\leq 10\%$). Figure 4.6 shows the number of the guests that get their spots denied at the restaurants.

Figure 4.6: Count guest spots denied over time per restaurant. Average of 20 repetitions with smoothed function, baseline scenario

We can see here that in general the line for all the three restaurants is zero (overlapping), meaning that no spots get denied. The only spike is for restaurant two in the first week (on Monday and Thursday). Even though the number goes up to around

60 which means a percentage of more than 10%, we do not see this as a thread to the pattern. The graph clearly shows that after the first week, the amount of spots denied goes down to zero. Furthermore, this only happens for restaurant two and no other restaurant.

The reasons for this behavior are twofold. The first one is related to the guest characteristic of having a high social dining tendency (see Table 4.3). This means that guests with this characteristic have a higher threshold for belonging and pleasure, and consequently go out more often. Furthermore, those guests also change their restaurant permanently if they got their spot denied at least two times [1]. This is proven by the data although not plotted). The data shows that same number of guests got their spot denied the first and the second time. Consequently, they changed their restaurant afterwards. This is further emphasized by size of the restaurant (see Table 4.8). Restaurant two is the smallest restaurant with only 78 seats, while having the same number of guests as the other two larger restaurants. This results in larger number of guests getting their spots denied initially, even though the guests with a high social dining tendency are only a subgroup of the guests of restaurant two.

The second reason is that agents do not have a typical behavior pre-defined. They learn over the course of the simulation. This means that at the beginning when the leisure need is still satisfied, and the pleasure and belonging needs are only slightly urgent, they will immediately go out, such as on the first Monday evening of the simulation. This can loosely be interpreted as moving to a new place, where the person also does not know what typical days to go out are, or which times are the best and so forth.

Taking these paragraphs above together, we argue that the final pattern for the guest agents is also met and the spikes at the beginning do not pose a thread to pattern of less than 10% spots denied. Since we have successfully proven now all patterns of the guest agents for the expectations, we move on now and look at the results of the restaurant owners in the next section.

4.6.1.2 Restaurant Owner Results

The part of Expectation 1 which focuses on the restaurant owner is the second part, namely *The restaurants are profitable, and the owners can run the restaurant in the way they want to*. Here again, we can split it into two patterns: 1) the restaurant is in a healthy financial situation (≥ break-even), and 2) the owners can run the restaurant in the way they want to (need depending).

1) Healthy financial situation: A healthy financial situation means that each restaurant is profitable. We can see that this is indeed the case. Figure 4.7 shows that all three restaurants are doing financially well and make money.

The reason why restaurant one is higher than the other two restaurants is that it is the largest restaurant and makes more profit per dish sold compared to the other two restaurants. The higher profit comes from the lower costs (profit = income - costs), as they own the location of their restaurant (see Section 4.3.1.2). Right at the end of the graph, it seems that there is a turning point, but that is not the case. Since the capital

[1] See here the restaurant changing mechanic

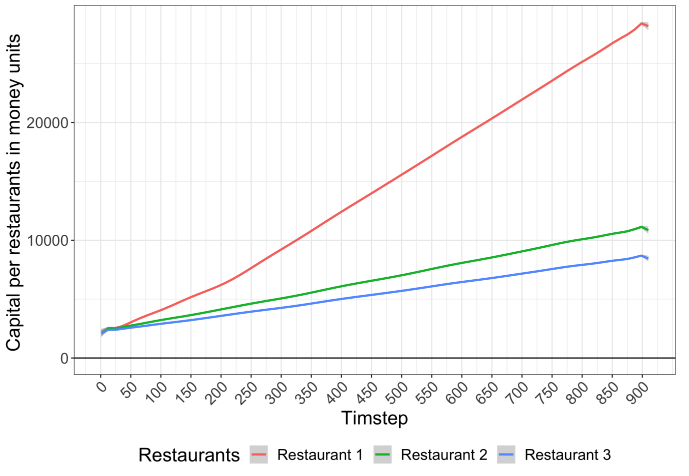


Figure 4.7: Capital per restaurant owner over time. Average of 20 repetitions with smoothed function, baseline scenario

is steadily increasing, it is fair to assume that if the simulation would have run longer, that the capital would keep doing so. Consequently, we see this part of the expectations as proven.

2) Running the restaurant they way they want to: To determine what we mean by 'the way they want to', we can look at the description of the restaurant owners (Section 4.3.1.2). For restaurant owner one, we can see that the quality of ingredients (the self-realization need) plays a very important role and the relationship to their regular customers (security need). For restaurant owner two money (the financial stability need) is very important and for restaurant owner three it is both money (the financial stability need), and good customer relations to their regular customers (security need).

With this in mind, we can look now at Figure 4.8 below. It shows that all needs for all restaurant owners are well-satisfied at the maximum level of 1 (the lines for each need are overlapping here). Consequently, we see this part of the expectations as proven.

4.6.1.3 Conclusion

In this scenario, we investigated our baseline scenario, where the size-based restriction norm is not active. We formulated our expectations for this scenario in Expectation 1: *The needs of the guests and restaurant owners are well satisfied. The guests go to the restaurant between 0.5 (once every two weeks) to twice per week. Only a few spots are denied (≤ 10%). The restaurants are profitable, and the restaurant owners can the restaurant the way they want to*. Our presented simulation results meet the required patterns, and consequently, we see Expectation 1 as met, see. Table 4.10 for a summary. This means that the scenario confirms the ability of our proposed agent

Figure 4.8: Average needs over all restaurant owners over time, all satisfied at the maximum level of 1.0. Average of 20 repetitions with smoothed function, baseline scenario

deliberation is able to exhibit plausible behavior in the case that the norm is not active and is therefore suitable for further investigation.

Table 4.10: Summary to check if the required patterns have been met

Required Pattern	**Is the pattern met?**
Guest needs well satisfied	✓, Guest needs stable and above threshold, Figures 4.2 and 4.3
Guests go to restaurant between 0.5 to 2 times per week	✓, Guests go to restaurant on average between 0.5 (once every two weeks) and 2 times per week, Figures 4.4 and 4.5
Only few spots denied	✓, No spots denied, only in the first during the learning phase, Figure 4.6
Restaurant owner needs well satisfied	✓, Restaurant owner needs stable at maximum need satisfaction possible, Figure 4.8
Restaurants are profitable	✓, Restaurants are making steady profit, Figure 4.7

4.6.2 Scenario 2 - Norm as limitation (only obedience)

In this scenario, the factor that we are changing is the norm. The norm will be active after nine weeks (a third of the total simulation run). The actions for the agents are fixed, i.e. the agents have *only* the same actions as in scenario one. This implies the agents are *limited* by the norm, as they can only obey the norm, and cannot take any actions against it. Our expectations for this scenario are as follows:

Expectation 2 (Norm as limitation)
The guests that are not affected by the norm (regular guests) do not alter their behavior. The guests that are affected by the norm (non-regular guests) alter their behavior as another already existing alternative action (going to the park) is more attractive now. More guests get their spots denied compared to the previous scenario. The restaurants go broke, and do not recover form that, as the restaurant owners can only abide the norm.

The distinction between the regular guests and non-regular guests is important here, as regular guests are not affected by the norm since they feel their spot at the restaurant to be secure. Contrary to this, the non-regular guests are affected by norm, as they do not believe that they have their spot secure at the restaurant, see the guest characteristics in Table 4.3. This means that the security[1] need is depleting over time for the non-regular guests once the size-based restriction norm becomes active. This is not the case for the regular guests.

Looking at them together, the regular guests and the non-regular guests are two sides of the same coin. One behavior cannot explained without the behavior of the other, as they share the same physical spaces. However, it is also important to look at each side individually, as the required patterns focus also on behavior comparison before and after the norm was active within one group of guests (e.g. non-regular guests change their behavior after the size-based restriction norm became active compared to before the norm was active) and not only between different groups. This is why we look at the two different guest groups, regular guests and non-regular guests, separately. When the behavior of one group is influence the behavior of another group, we will explain it and refer to it.

4.6.2.1 Guest Results

The parts of Expectation 2 which relate to the guest agents are as follows: *The guests that are not affected by the norm (regular guests) do not alter their behavior. The guests that affected by the norm (non-regular guests) alter their behavior as another already existing alternative action (going to the park) is more attractive now. More guests get their spots denied compared to the previous scenario*. To substantiate this, the model needs to produce the following four patterns: 2a) Needs of the guests are satisfied ($\geq$ threshold)[2], 2b) guests not affected by the norm (regular guests) do not alter their behavior, 2c) guests affected by the norm (non-regular guests) go to the park instead of the restaurant, and 2d) more spots are denied compared to the baseline scenario.

2a. All need satisfied: This can be observed in Figures 4.9 and 4.10 as they show the average needs satisfaction over time for the regular guests (Figure 4.9) and non-regular guests (Figure 4.10).

[1]The need related to feeling secure to have a place at the restaurant and park

[2]Not mentioned directly in the expectation, but the needs are at the core of the decision-making process, and are thus always in the focus of the analysis.

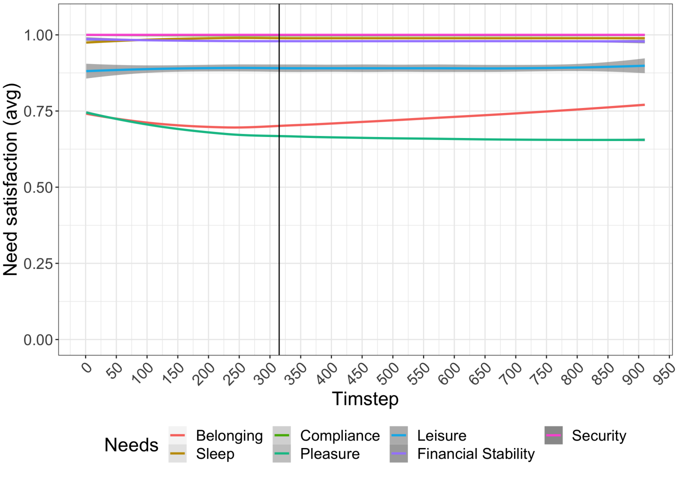


Figure 4.9: Average need satisfaction over all regular guests over time. The black vertical line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm as limitation scenario

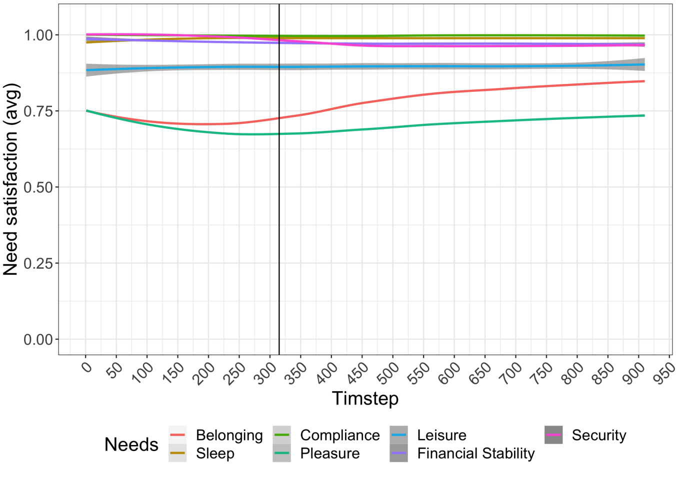


Figure 4.10: Average need satisfaction over all non-regular guests over time. The black vertical line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm as limitation scenario

When comparing Figures 4.9 and 4.10 with Figure 4.2, we can see that the need satisfaction of the regular guests (Figure 4.9) is identical with the need satisfaction of all guests from the previous scenario (Figure 4.2), whereas the one from the non-regular guests (Figure 4.10) differs in terms of pleasure, belonging, and security. This further emphasizes that the non-regular guests adapted their behavior, and that the regular guests kept their behavior identical as in the scenario where the norm is not active.

The pleasure, and belonging need satisfactions of the non-regular guests are higher compared to the regular guests, while the security satisfaction is lower, as shown in Figure 4.11. The reason for this difference is that the non-regular guests are in a state of uncertainty after the norm became active, meaning that their security need is depleting. To satisfy this need again the guests go out which in turn then also satisfies their belonging and pleasure need more. While this is a side-effect from the design of the model, it is also not far from reality. The non-regular guests feel not safe, so they go out just in case the next time they want to go out, they do not get a spot.

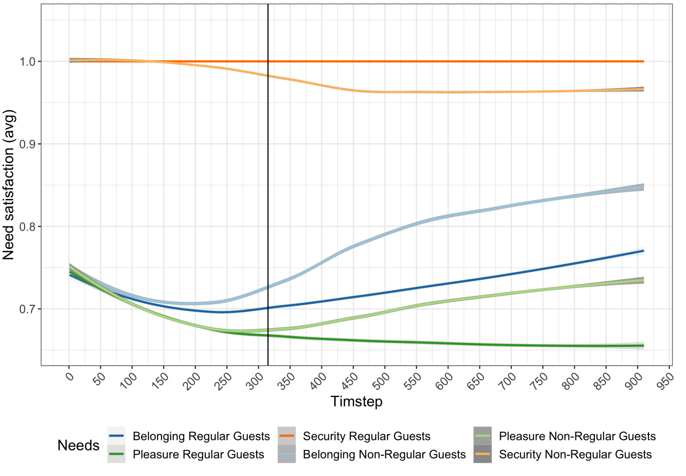


Figure 4.11: Comparison of the average need satisfaction of the belonging, pleasure, and security needs, between the regular and non-regular guests. Average of 20 repetitions with smoothed function, Norm as limitation scenario

The last part to address the first pattern can be seen in Figures 4.12 and 4.13 showing the difference between the average need satisfaction of the regular guests (Figure 4.12) and the non-regular guests (Figure 4.13) to their respective thresholds.

These figures show that the needs are stable and well satisfied, meaning that they are above the threshold line. The reason that in both figures the financial stability need is a bit below zero, is because of the aforementioned smoothed averages. This is also the case for the pleasure need in Figure 4.12. But since they are close to the zero line. We consider this no problem to the targeted pattern of the guest agents satisfying their needs. Furthermore, we can see that the y-axis scale is very small (numbers close to

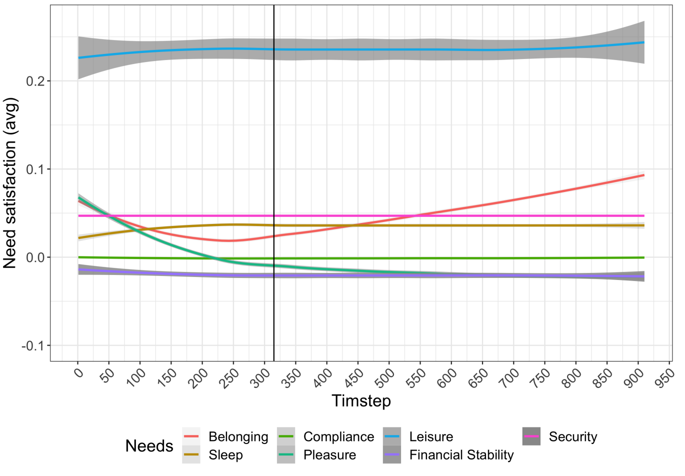


Figure 4.12: Average difference between the need satisfaction of the regular guests and their respective thresholds, value calculation: current fill level - threshold. Positive value: Need satisfied (above threshold. Negative value: Need not satisfied (below threshold). Average of 20 repetitions with smoothed function, Norm as limitation scenario

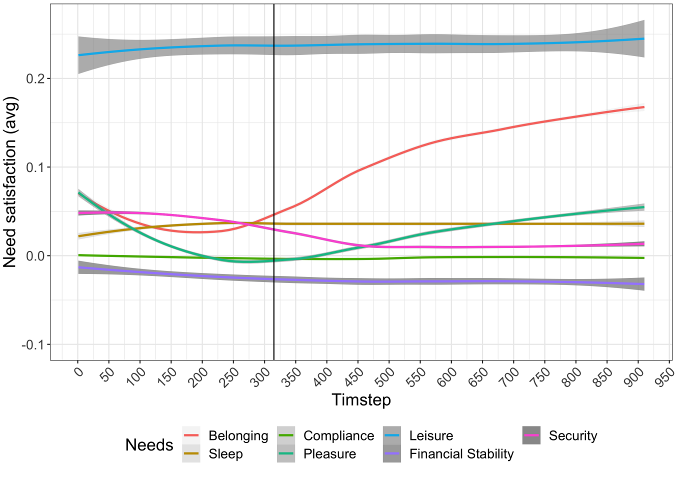


Figure 4.13: Average difference between the need satisfaction of the non-regular guests and their respective thresholds, value calculation: current fill level - threshold. Positive value: Need satisfied (above threshold. Negative value: Need not satisfied (below threshold). Average of 20 repetitions with smoothed function, Norm as limitation scenario

zeros. This means that the difference is only very minor and basically at the zero line.

Furthermore, Figure 4.13 shows the drop in security need satisfaction when the size-based restriction norm became active, together with the increase in belonging and pleasure need satisfaction.

Taken everything together, we conclude that the pattern *all guest needs (regular guests and non-regular guests)* of the expectation is successfully addressed.

2b. Unadapted regular guest: To demonstrate that the regular guests do not change their behavior to satisfy their needs, we show that they go just as often do the restaurant in scenario two as in scenario one. This means that they keep going to the restaurant like before the norm became active (as indicated by the stable trendline), and do not prefer the park over the restaurant. Figures 4.14 and 4.15 show the amount of regular guests at the restaurant (Figure 4.14), and park (Figure 4.15) over time respectively.

Figure 4.14: Count of regular guests at restaurant over time. The black horizontal lines show the seat limit for each restaurant before and after the norm activation. The black vertical lines shows when the norm was activated. The trend line is lower than the heights, as many days have the value zero. Average of 20 repetitions with smoothed function, Norm as limitation scenario

The figures show that the regular guest's behavior remains stable before and after the norm became active (as indicated by the vertical back line), confirming the expected pattern. Further confirmation for this pattern can be found in Figure 4.16. The boxplots show the average times a regular guest goes to the restaurant (left side boxplots), and park (right side boxplots) per week. Again the figure shows that there is no difference between the left sides in the graphs before the norm activation and the right side of the graph (restaurant two see the next paragraph), after the norm became active. This further indicates that the regular guests are not changing their behavior.

The behavior of the regular guests of restaurant 2 stands out as a potential threat

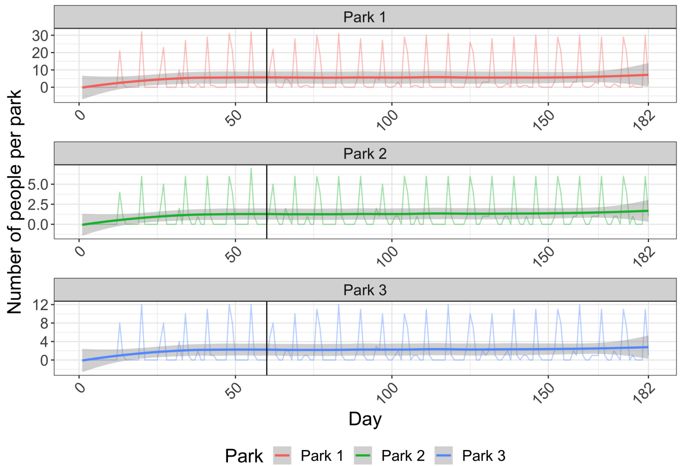


Figure 4.15: Count of regular guests at the park over time. The black vertical line indicates the moment when the norm was activated. The trend line is lower than the heights, as many days have the value zero Average of 20 repetitions with smoothed function, Norm as limitation scenario

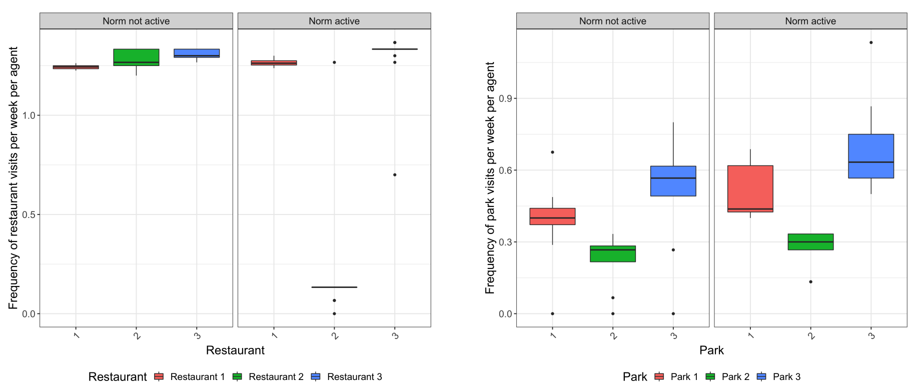


Figure 4.16: Average visit regular guests to the restaurant (left) and park (right) per week, value calculation: total number of regular guests per restaurant (left) and park (right) per week divided by the regular guest agents assigned to that restaurant/park. Average of 20 repetitions, Norm as limitation scenario

to the targeted pattern of unadapted regular guests, and as such we want to discuss it now. It seems that hardly any regular guests from restaurant two stay in restaurant two. The raw data (obscured by the trend line but visible in the box plot) show that a small number of regulars still choose restaurant two, but many switch to restaurant three after being denied spots at restaurant two (see the green spike in Figure 4.20 after the norm became active).

The immediate switch after one time spot denied is due to the connection of the regular guests with their restaurant. As we described in the guest agent make-up (see Section 4.3.1.1) and restaurant owner agent make-up (see Section 4.3.1.2), regular guests are valued by the restaurant owner, and thus when they get their spot denied once, they immediately change their restaurant, as their loyalty to the restaurant is betrayed. This is a decision that we made for the implementation to reflect the connection regular guests can form with their restaurant. The reason they switch to restaurant three is due to the numerical order of switching the restaurants, as mentioned in Section 4.3.1.1. Guests from restaurant one first try restaurant two, and then restaurant three. Guests from restaurant two first try restaurant three and then restaurant one, and guests from restaurant three first try restaurant one, and then restaurant two.

Taking all the discussions above together, we see the pattern of the regular guests not adapting their behavior as proven. With not adapting behavior here we mean keep going to the restaurant and not the preferring the park. Thus, the regular that change their restaurant are exempted in this sense, as they are still going to the restaurant.

This pattern is also strongly influenced by the behavior of the non-regular guests. Since they prefer going to the park and do not go to the restaurant anymore after the norm becomes active, it opens up the space for the regular guest to go to the restaurant, as more space is available now to them. We discuss the behavior of the non-regular guests and their corresponding pattern next.

3. Non-regular guest behavior: In contrast to the regular guests, we expect the non-regular guests to adapt their behavior, and prefer a different action and therefore go to the park instead of the restaurant, as they perceive that their spot is guaranteed there[1]. Looking at Figures 4.17 and 4.18, we can see that after the size-based restriction norm became active, the majority of non-regular guests goes to the parks and not to the restaurants anymore (as indicated by the rising and falling trendlines respectively) which then in turned opened up the space for the regular guests at the restaurants.

Figure 4.19 further confirms the targeted pattern of non-regular guests adapting their behavior. The boxplots show the average visit per week of the non-regular guest agents for each restaurant and park respectively. It can clearly be seen that before the norm was active, the people went more often per week to the restaurant, and now after the norm is active, the people go to the park more often. Taking everything together, we argue that the desired pattern for the non-regular guests, i.e. the non-regular guests adapting to the size-based restriction norm and thereby preferring the park over the restaurant to satisfy their needs, is confirmed.

As mentioned earlier, these discussed behavioral results of the non-regular guests together show, why they are the other side of the coin of the regular guests. Since the

[1]we assume that the park has always space, and thus guarantees a spot.

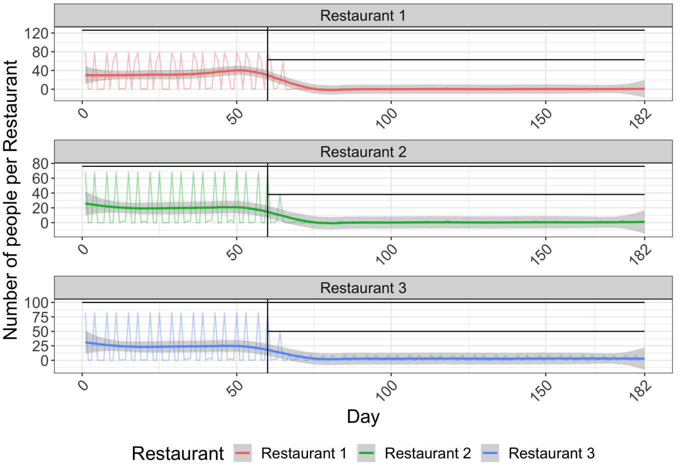


Figure 4.17: Count of non-regular guests at restaurant over time. The black horizontal lines show the seat limit before and after the norm activation. The black vertical line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm as limitation scenario

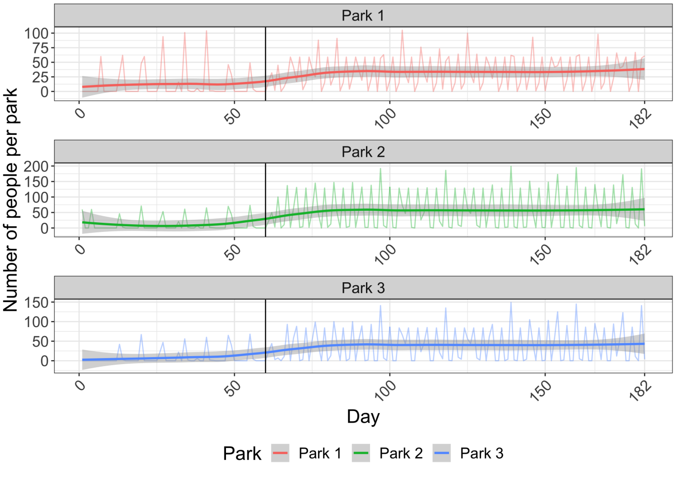


Figure 4.18: Count of non-regular guests at park over time. The black vertical lines show when the norm was activated. Average of 20 repetitions with smoothed function, Norm as limitation scenario

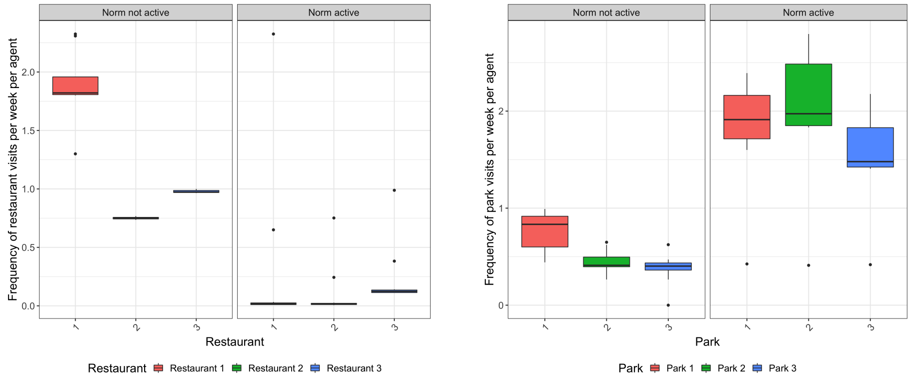


Figure 4.19: Average visit of non-regular guests to restaurant (left) and park (right) per week, value calculation: total number of non-regular guests per restaurant (left) and park (right) per week divided by the non-regular guest agents assigned to that restaurant/park. Average of 20 repetitions , Norm as limitation scenario

non-regular guests prefer the park over the restaurants, the regular guest have space available at the restaurant. This shows that our PBADA agent deliberation architecture is capable of displaying this complex intertwined behavior.

Increase in spots denied: This is the final pattern for scenario two. Figure 4.20 shows the number of people that get their spot denied for each restaurant. Similarly as in scenario one, we can see that before the norm gets activate the behavior is the same (see Figure 4.6), and thus, the arguments from scenario one hold here as well.

After norm activation (black vertical) we can see a big green spike, medium red spike, and a small blue spike. The small blue spike represents the spots denied in restaurant three. The medium spike represents the number of spots denied in restaurant one. The big green spike represents the spots denied in restaurant two. The reason for these various spikes can be found in the needs and the restaurant changing mechanic.

The spots are denied immediately after the size-based restriction norm becomes active. At this stage, the security need satisfaction of the non-regular guests has started to decrease, as they can now not be sure anymore that they have their spot available. However, their need is still satisfied so early after the size-based restriction norm has been activated, and thus not depleted enough that it requires an action to be satisfied again.

As a result of this, on the first day of the size-based restriction, some non-regular guest agents act basically as if the norm is not active yet (see here also the restaurant visits in Figure 4.17) . Furthermore, restaurant one has a large amount of regular guests (see Table 4.8). This means that the guests from restaurant one keep going to the restaurant. Furthermore, if the regular and non-regular guests get their spot denied and have a high social drive, they go immediately to restaurant two in the same timestep. This is then reason why for restaurant two the spike is so large, as it is now not only the people from restaurant two that keep going there, but also the people from

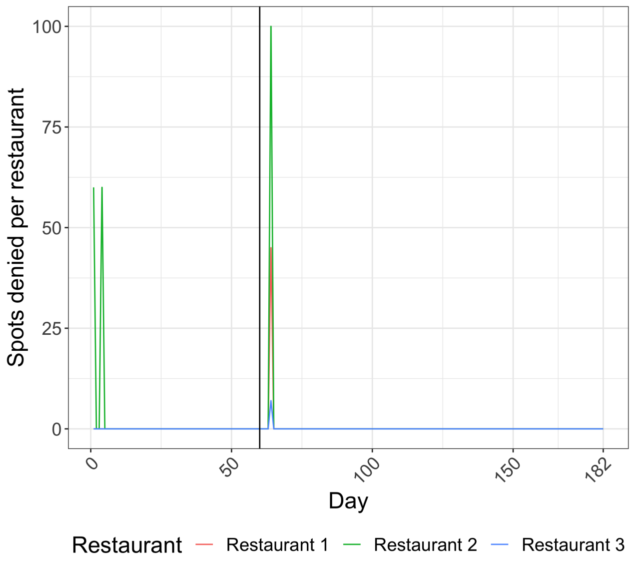


Figure 4.20: Count of spots denied over time per restaurant. The black vertical line shows when the norm was activated, Norm as limitation scenario. Average of 20 repetitions with smoothed function, Norm as limitation scenario

restaurant one which got their spot denied. This is further emphasized by size of the restaurant (see Table 4.8). Restaurant two is the smallest restaurant with only 78 seats. This means that when the size-based restriction norm is active, restaurant two has only 39 seats available. Finally, the spike of restaurant three is then rather small, because the guests (left) of restaurant two can have a lower social drive (as the ones with a higher social drive already changed at the beginning of the simulation when they got their spot denied, see argumentation for the previous scenario), and thus do not try the same day, but rather go home and try another day.

This behavior is not far from reality and not all people change their behavior immediately after a (new) norm becomes active. Some people stay reactive and continue their behavior as before. This is the case at this point in time for these agents, as their security need is still satisfied. Consequently, we see this last required pattern for the guests as met. This is further strengthened by the results, as no spots get denied after the spikes at the beginning of the simulation (as covered in the previous scenario), and the spike after the norm became active.

4.6.2.2 Restaurant Owner Results

The part of Expectation 2 that concern the restaurant owners is: *The restaurants go broke and do not recover from that, as the restaurant owners can only abide the norm.*. Due to the limited amount of guests that can come to the restaurant, and the limited action possibilities of the restaurant owners, we expect that they will simply go broke and do not recover from that. The desired pattern for this case is that the restaurant is doing financially well before the norm is active, and possibly goes broke after the norm becomes active. Furthermore, they will stay broke and do not recover from that. While this is to be expected given the model (and also mentioned in the verification in Section 4.5), it is also a pattern to be observed in reality. If the restaurant owner does not do anything, nothing will change, and the they will go broke.

Figure 4.21 shows that all restaurant owners are doing financially well before the norm, and are losing money after the size-based restriction norm becomes active (indicated by the black vertical line) and are going broke eventually. The varying time of them going broke is due to the introduced restaurant owner characteristics in Section 4.3.1.2. All restaurants going broke is matching the desired pattern. It is important to note here that we kept the simulation running to show that the restaurants keep losing money, and do not recover later on in the simulation. When the restaurants reach the point of being broke, the restaurant owners would need to decide between closing their restaurant or taking a loan from a bank or other place to keep their restaurant running. This however, goes too far to the purpose of our model to validate our proposed PBADA agent deliberation architecture, and is thus not modeled. Consequently, restaurants keep running and stay open for the guests to visit them.

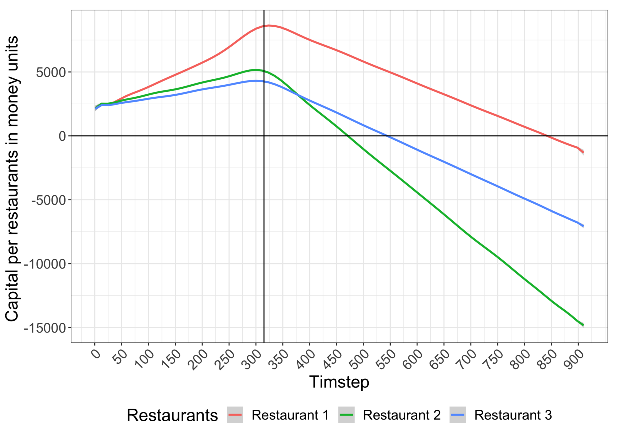


Figure 4.21: Capital for each restaurant owner over time. The black line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm as limitation scenario

While we have now proven this part of our expectations, we still want to look at how this affects the needs of the restaurant owner. The reason we are doing this is to show that the needs of the restaurant owner are not satisfied anymore. While this is something that we ensured through the verification process, it is still worth discussing here when comparing it to reality. In reality people would do something to satisfy their needs. This shows that seeing the norm only as a limitation on the agents behavior (people can only obey the norm) does not only miss potential behavior [82], but is also far from reality.

With the restaurants going broke, we can see in Figure 4.22 that the need satisfaction for the financial stability need of the restaurant owners is going down as well. The decline is step-wise as the restaurant owner checks their monetary gains/losses at the start of the new week by comparing their current amount of capital with that at the start of the previous week to identify if they gained money or lost money in the now finished week. The big drop in the beginning occurs, because when the norm becomes active and the number of available places is set to 50% of the total capacity of each restaurant (due to the size-based restriction norm), the financial stability need of each restaurant owner also drops to 0.5. This was a model specific choice. The graph shows a slightly higher number hidden behind the black norm activation line, as the restaurant owners still made profit that week. This is why in the following decline of the financial stability need, the line is not at 0.5 at the beginning of the decline.

The reason for the big steps in the beginning is that money (related to the financial stability need) is very important for restaurant owner two and three (see the description of the restaurant owners in Section 4.3.1). Thus, losing money results in a larger need satisfaction loss, thereby outweighing the smaller need satisfaction loss of restaurant owner one, for whom money is not that important.

The reason for the small steps at the end is that at this point in the simulation, restaurants two and three are already broke, and only restaurant one has still some money that it is losing. This means that the small steps are representing the loss of satisfaction of the financial stability need of restaurant owner one, as the need is already full depleted down to zero for restaurant owners two and three.

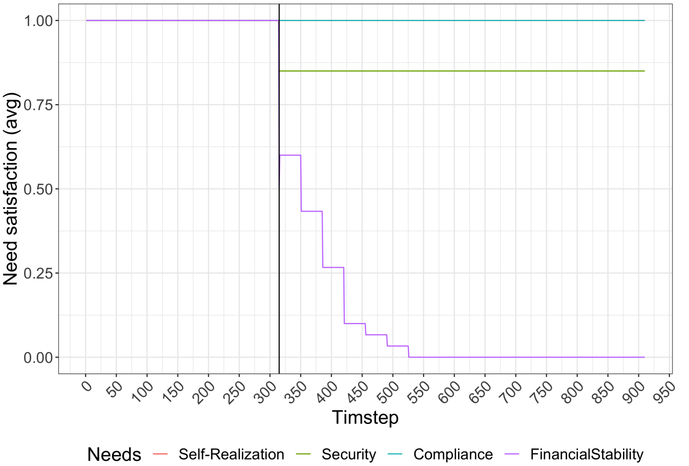


Figure 4.22: Average need satisfaction over all restaurant owners over time. The black vertical line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm as limitation scenario

Another thing we can see in Figure 4.22 is that the security need (green line) of the restaurant owners is dropping to a lower level after the norm became active, hidden behind the black vertical line showing that the size-based restriction norm became active. This is because there is a risk now that the restaurant owners have to deny spots to their regular customers and thus cannot take care of their customer relations to their full extend. The drop in the security need symbolizes this limitation.

4.6.2.3 Conclusion

In this scenario, we addressed the case where the norm is only seen as a limitation on the agent's behavior, meaning that the size-based restriction norm is active but the agents can only obey the norm without ways to counteract the perceived negative consequences of the norm, or violate it. Our expectations for this case, Expectation 2, were as follows: *The guests that are not affected by the norm (regular guests) do not alter their behavior. The guests that are affected by the norm (non-regular guests) alter their behavior as another already existing alternative action (going to the park) is more attractive now. More guests get their spots denied compared to the previous scenario. The restaurants go broke, and do not recover form that, as the restaurant owners can only abide the norm.*

To address these expectations, we firstly introduced the norm after a third of the simulation run, i.e. nine weeks, (difference to the previous scenario), and secondly kept the action space of the agents the same (same as in the previous scenario). Table 4.11 shows that we successfully addressed all required patterns for the expectation, and thus, we conclude that we successfully met the expectation for this scenario.

Table 4.11: Summary to check if the required patterns have been met for the norm as limitation scenario

Required Pattern	**Is the pattern met?**
Guest needs well satisfied	✓, Guest needs stable and above/around threshold, Figures 4.9, 4.10, 4.12 and 4.13
Guests not affected by the norm (regular guests) do not alter their behavior	✓, Regular guests go in the same manner to the restaurant and park before and after the norm became active, Figures 4.14 to 4.16
Guests affected by the norm (non-regular guests) alter their behavior	✓, Non-regular guests prefer the park over the restaurant after the norm became active, Figures 4.17 to 4.19
More spots denied compared to the previous scenario	✓, More spots are denied after the norm became active, Figure 4.20
Restaurants go broke, and do not recover	✓, Restaurants lose money, Figure 4.21

Comparing the results from scenario two with the results from scenario one, where the norm is not active, we can see that the norm does impact the behavior of the agents significantly, and a change of behavior of the agents compared to the previous scenario can be seen for the non-regular guest agents. It might seem that the guests actively deliberate about the norm and decide to go the park. While this is true to some extend (as they move one layer down in terms of the complexity of the deliberation, namely to the medium-complex deliberation level), it does not capture the complete picture, and cover all potential reactions. Furthermore, it is clear that the restaurant owners would do something to not go broke.

4.6.3 Scenario 3 - Norm Response Diversity

For the final this scenario, we change the factor of the available actions for the agents. The agents have now an increased action space, meaning more actions available to them. The guests can now reserve a spot at the restaurant, and the restaurant owners can either violate the norm, or counteract the norm by using cheaper ingredients, see Tables 4.4 and 4.7.

To respond to this potential behavior by the restaurant owner, the characteristic of the guest agents of having a sensitive taste (see Table 4.3 now comes into play. Actions do not happen in a vacuum. Agents (people) will respond to the actions taken by other agents (people). In our case, this means that some guests will respond to the actions taken by the restaurant owner by getting less pleasure from going to the restaurant and potentially prefer going the park over the restaurant, if the restaurant owner decides to change their quality of ingredients. Similar to the approach taken for the regular and non-regular guests in the previous scenario, we will look at the behavior of those guest groups (have sensitive taste and not have sensitive taste) separately,

Furthermore, the norm will again be active after nine weeks in the simulation run, like in the previous scenario. Our expectations for this scenario are as follows, Expectation 3:

Expectation 3 (Norm response diversity)
The restaurant owners will violate the norm by letting in more guests than allowed and/or use cheaper ingredients to increase their profit. The restaurants are profitable. The guests not affected by the norm (regular guests) do not alter their behavior. The guests affected by the norm (non-regular guests) reserve a spot at the restaurant. The guests affected by the ingredient change (having a sensitive taste) react to it and go to the park.

4.6.3.1 Guest Results

The parts of this expectations relevant for the guest agents are: *The guests not affected by the norm (regular guests) do not alter their behavior. The guests affected by the norm (non-regular guests) reserve a spot at the restaurant. The guests affected by the ingredient change (having a sensitive taste) react to it and go to park*[1]. To substantiate this part of the expectations, the guest agents need to substantiate the following patterns:

[1]Our expectation is the park here and not another restaurant, as we did not model the knowledge or possibility that guests have assumptions about the ingredients used by restaurants that they are currently not visiting, as it is not relevant for the validation of our PBADA architecture, and thus out of the scope for this model for this thesis.

1) The needs of the guests are well satisfied (≥ threshold)[1], 2) people affected by the change of the quality of ingredients will stop going to the restaurant and go to the park instead[2], and 3) non-regular guests are reserving a spot at the restaurant if they want to go there.

1. Guest need satisfaction: When looking at the results for the guest agents, we can see in Figure 4.23 that the needs are overall satisfied. Figure 4.24 confirms this, as this graph shows the differences between need satisfaction over time and the threshold of each respective need respectively. All needs, with the exception of pleasure and financial stability, are satisfied above their respective threshold, meaning that the lines are evolving around the y-axis value of zero which indicates that the satisfaction level of the need has exactly the same value as the threshold of the respective need. The reason that those two needs seem not to be satisfied is due to the smoothing, as explained for example in Section 4.6.1, where the situation was similar. The needs are satisfied but to the applied smoothing function they do not seem to be as for some agents the needs might not be satisfied at this point in time of the simulation. But for many agents it is. Furthermore, they are very close to the zero line, as shown by the y-axis scale in Figure 4.24. Consequently, we do conclude that the pattern that all guest needs are satisfied is met.

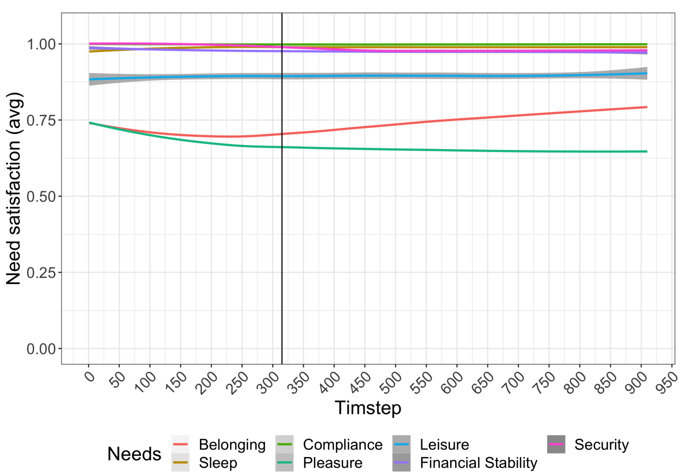


Figure 4.23: Average need satisfaction over all guest agents over time. The black vertical line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm response diversity scenario

The reason that we did not make separate need graphs (for all needs) for the guests that have a sensitive taste, and the ones that do not have a sensitive taste, as we we

[1]Not mentioned directly in the expectation, but it is always with the focus on adapting behavior to satisfy one's needs

[2]Some restaurant owners in this scenario will use cheaper ingredients resulting in less pleasure for the guests. Consequently, this reaction needs to be part of the patterns to show that our proposed agent deliberation architecture shows plausible behavior.

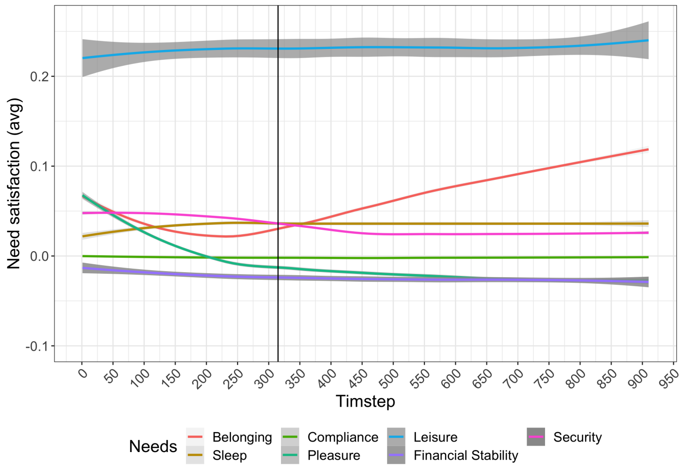


Figure 4.24: Guest need threshold difference over time. The black vertical line shows when the norm was activated, value calculation: current fill level - threshold. Positive value: Need satisfied (above threshold). Negative value: Need not satisfied (below threshold). Average of 20 repetitions with smoothed function, Norm response diversity scenario

did for the regular and non-regular guests in the previous scenario, can be found in Figure 4.25.

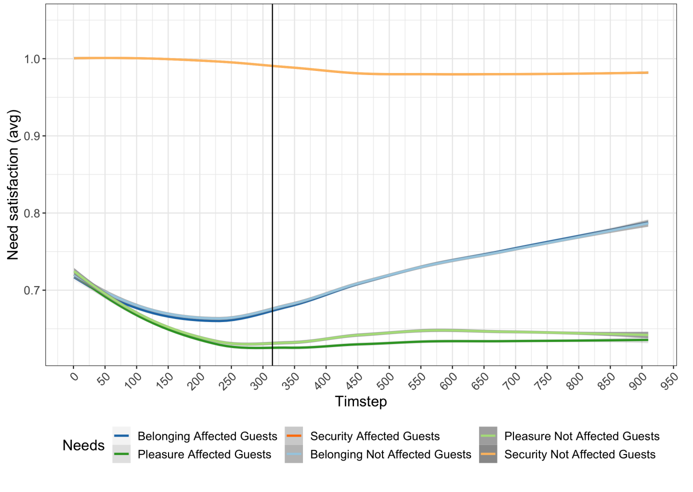


Figure 4.25: Comparison of the average need satisfaction of the belonging, pleasure, and security needs, between the regular and non-regular guests. Average of 20 repetitions with smoothed function, Norm response diversity scenario

Figure 4.25 shows the difference in the belonging, pleasure, and security needs for the guests affected by the change of quality in ingredients of the restaurant owner and the ones not affected by it. We can see in the figure that the belonging and security need are equally satisfied, as both lines for guests affected and not affected by the ingredient change are overlapping. The only difference can be found in the pleasure need with the pleasure need satisfaction of the guests affected by the ingredient change being slightly lower compared to the pleasure need satisfaction of the guests not affected by the ingredient change. However, the difference is only very small.

The reasons are twofold. The first one is that restaurant owner one will not use cheaper ingredients compared to restaurant owners two and three[1], and thus, the guests that have a sensitive taste and also go to restaurant one, have no negative impact on their pleasure need. The guests of the other two restaurants that have a sensitive taste, go to the park, as they get a higher pleasure need satisfaction compared to going to the restaurant when the restaurant owner is using cheaper ingredients, but get a lower pleasure need satisfaction gain from going to the park when the restaurant owner is using high quality ingredients (the starting case in the simulation).

Before moving to the second pattern to show how the need satisfactions of the agents come about through their behavior, we want to mention that Figure 4.23 looks similar to Figure 4.2 from the scenario where the norm is not active and Figure 4.9 from the regular guests of scenario two. This shows that the behavior exhibited by the agents in this scenario is similar to their desired behavior if the norm was not active (see the results of the first scenario where the norm was not active), and thus necessary to have all possible norm reactions available. This is different than for the non-regular guests in previous scenario (see Figure 4.10) where the guests were not able to reserve their spot.

2. Guests that have a sensitive taste do not go to the restaurant anymore: To address this pattern, we look at the behavior of the guests that are affected by the ingredient change, and the ones that are not affected by the ingredient change, as they are each others inverse patterns, similar to the case of the regular and non-regular guests of the previous scenario.

Figures 4.26 and 4.27 show the behavior of the guest agents that have a sensitive taste, and are thus affected by change of ingredient quality by the their respective restaurant. Figure 4.26 shows the amount of people at the restaurant over time, and Figure 4.27 shows the amount of people at the park over time. We can see in these graphs that the behavior of the guest agents for restaurant one does not change, as restaurant owner one is not changing the quality of their ingredients[2]. Furthermore, we confirm that due to the use of lower quality ingredients, the guests of restaurant two and three prefer the park over the restaurant, seen by the downtrend of people going to the restaurant after the norm has been activated (the black vertical line).

Similarly to scenario two, we can now look at Figure 4.28 showing the box plots of the average restaurant visit (left) and park visit (right) for an agent being affected by the ingredient change (sensitive taste). Each plot is separated in two parts, the left one for the time period when the norm is not active, and the right one for when the

[1]Shown later in this scenario

[2]We show this in the later part of this scenario but have to mention it here, as it influences the results

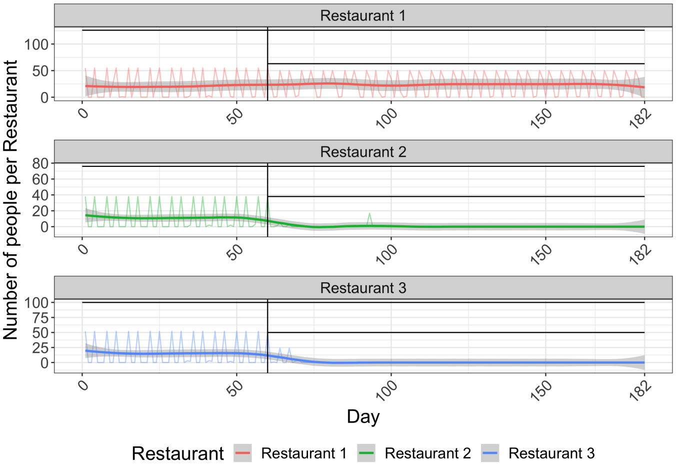


Figure 4.26: Count of the guests affected by the ingredient change (having a sensitive taste) at restaurant over time. The black horizontal lines show the seat limit before and after the norm activation. The black vertical lines shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm response diversity scenario

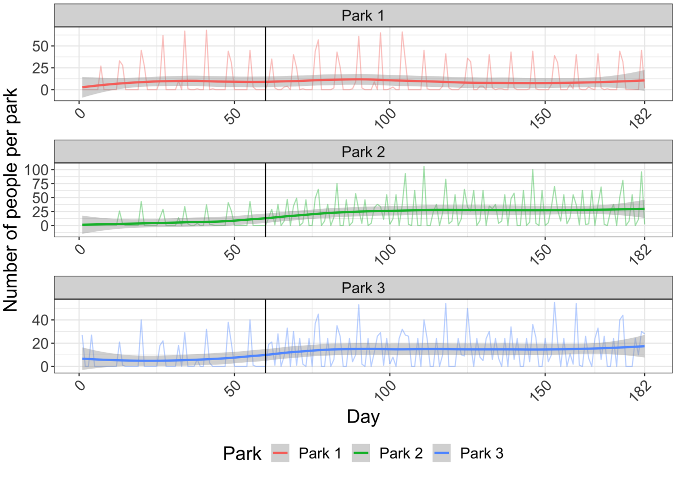


Figure 4.27: Count of guests affected by the ingredient change (not having a sensitive taste) at park over time. The black vertical line shows when the norm was activated. The trend line is lower than the heights, as many days have the value zero. Average of 20 repetitions with smoothed function, Norm response diversity scenario

norm is being active. We can see that the discussed pattern from above is confirmed with this plot. The small variations for restaurant one come from the fact that in this boxplot the regular and non-regular guests are mixed together, because the focus is on the guest agent characteristic of having a sensitive taste (i.e. being affected by the change of quality of ingredients by their respective restaurant owner, see Table 4.3). Thus, the same argument from the previous scenario holds, where some people go to the restaurant to satisfy their security need by reserving a spot in the restaurant[1].

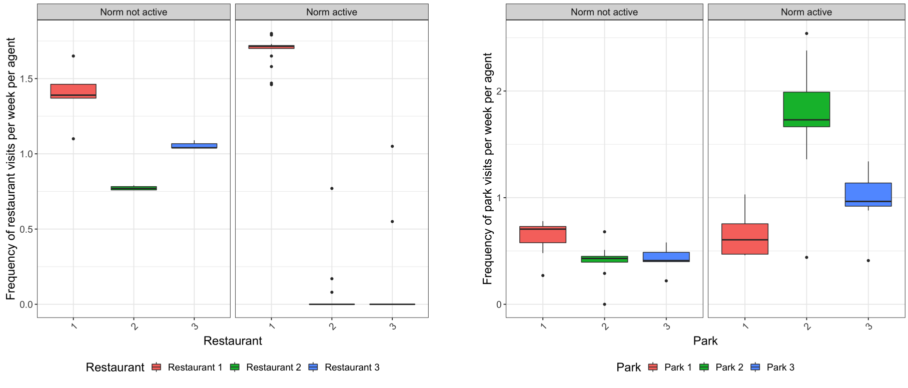


Figure 4.28: Average restaurant visits (left) an park visits (right) per week for a guest agent being affected by the ingredient change of the restaurant owner, value calculation: total number of guests that are affected by the ingredient change per restaurant (left) and park (right) per week divided by the guest agents that are affected by the ingredient change, and assigned to that restaurant/park. Average of 20 repetitions, Norm response diversity scenario

The key point here is that there can be a significant trend change observed for restaurants two and three from before the norm is active, and after the norm is active, as this is when the restaurant owners of restaurant two and three are using cheaper ingredients.

Figures 4.29 and 4.30 which show the behavior of the guest agents that do not have a sensitive taste, meaning that they are not affected by the ingredient change of their restaurant owner. These guests get the same amount of pleasure from going to the restaurant as before the ingredient change, because the food ’tastes the same’ for them. The behavior of these people does not change (see the trend line), and that they keep going to the restaurant like they did before the norm became active (the black vertical line).

Further confirmation of this can be found in Figure 4.31 below showing the boxplot of the average restaurant visit (left) and park visit (right) per week for an agent that is not affected by the ingredient change. We can see (for each side respectively) that there is no significant difference between the norm not being active (left) and being active (right). This means that even though restaurant owner two and three are using cheaper

[1]Shown later in the results.

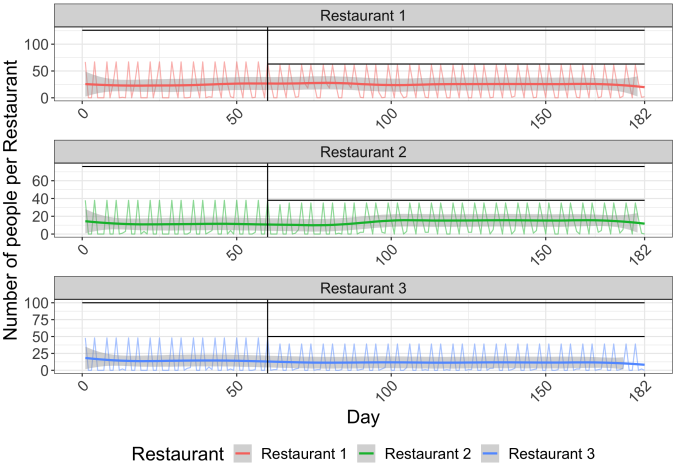


Figure 4.29: Count of the guests not affected by the ingredient change (not having a sensitive taste) at restaurant over time. The black horizontal lines show the seat limit before and after the norm activation. The black vertical line shows when the norm was activated. The trend line is lower than the heights, as many days have the value zero. Average of 20 repetitions with smoothed function, Norm response diversity scenario

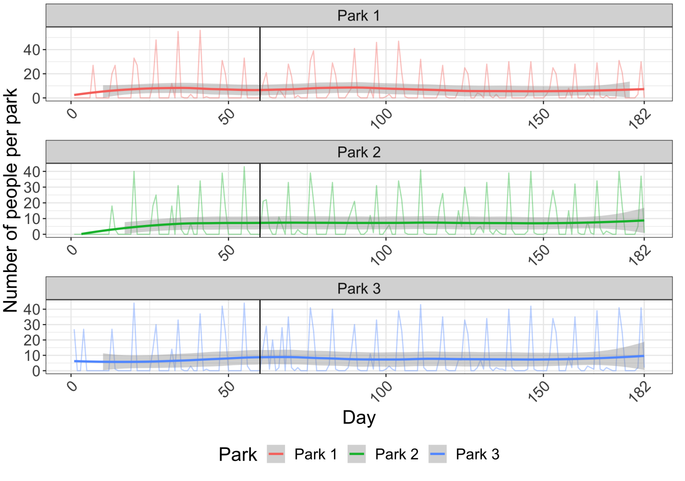


Figure 4.30: Guests not affected by the ingredient change (not having a sensitive taste) at park over time. The black vertical line shows when the norm was activated. The trend line is lower than the heights, as many days have the value zero. Average of 20 repetitions with smoothed function, Norm response diversity scenario

ingredients after the norm became active, the guests do not change their behavior, as they are not affected by that change in their pleasure need satisfaction gain. The small differences between the graphs on each side in the figure come from the security need, as in these boxes the regular guests and non-regular guests are mixed together. Thus non-regular guests might go out more often to satisfy their security need (see earlier discussions). Based on these results, we see the pattern as successfully proven.

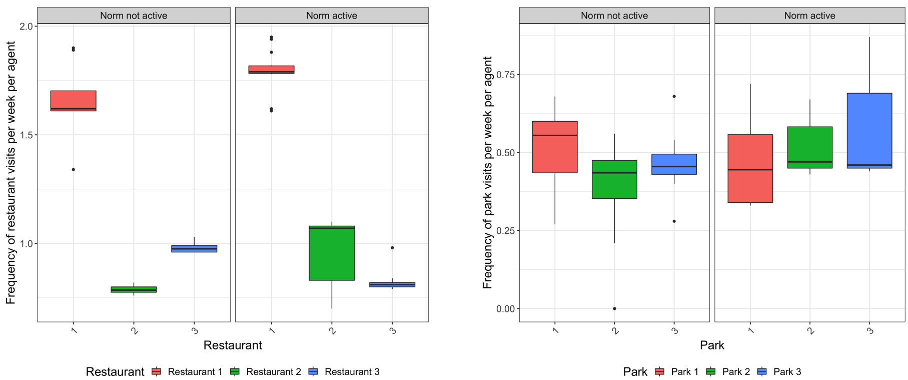


Figure 4.31: Average restaurant visits (left) an park visits (right) for a guest agent not being affected by the ingredient change of the restaurant owner, value calculation: total number of guests that are not affected by the ingredient change per restaurant (left) and park (right) per week divided by the guest agents that are not affected by the ingredient change, and assigned to that restaurant/park. Average of 20 repetitions, Norm response diversity scenario

3. Non-regular guests reserving a spot: The last pattern for the guests agents that we have to show is that the non-regular guests reserve a spot in their respective restaurant when they want to go to the restaurant. This is a difference to the previous scenarios, as the guests have now this action available which makes the restaurant more attractive than the park again.

To see if people reserve their spot, we can look at Figure 4.32 showing the total number the actions to go to the restaurant with reservation and to go to the restaurant without reservation were performed by the regular guests and non-regular guests respectively. When looking at this figure, we can see two things. The first one is that people do reserve their spots, as the action count for going to the restaurant with reservation is above zero. With this result we see the pattern as successfully proven.

Nonetheless, we want to talk about the large difference between the action count of going to the restaurant with reservation and without reservation. There are two main reasons fo that. The first one is that this bar chart mixes all agents together, and only provides a difference between regular and non-regular guests. Furthermore, this bar chart represents a total count over the whole simulation run. This means that the count for the action of not reserving a spot in the restaurant also includes the time before the size-based reaction was active. During the time before the norm became

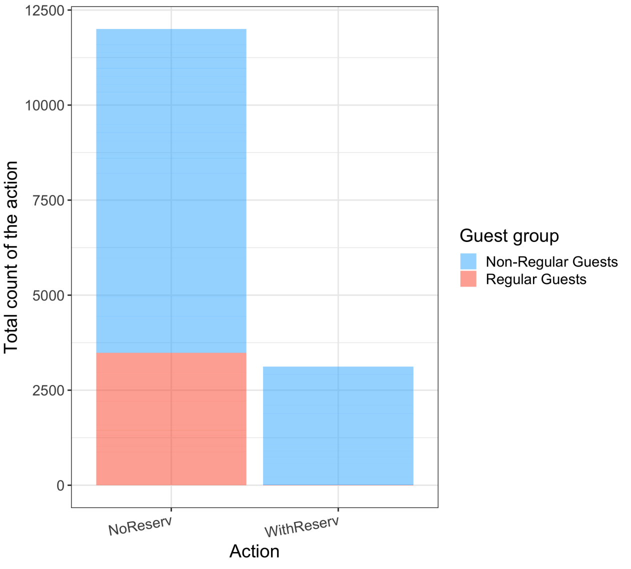


Figure 4.32: Total count of the actions of going to the restaurant *without* reservation and *with* reservation selected. Average of 20 repetitions with smoothed function, Norm response diversity scenario

active, the restaurants were doing financially well. and did not have to use cheaper ingredients. Furthermore, the non-regular guests also did not have to reserve a spot, as the size-based restriction norm was not active, and thus, their security need was not depleting. This means that the count for the action for not reserving a spot in the restaurant also includes the count over the all guests agents before the size-based restriction norm was active.

Effectively, the amount of people that reserve a spot in the restaurant are only the non-regular guests that are do not have a sensitive taste, and are thus counted by reserving a spot in the restaurant bar in Figure 4.32. This means only two out of eight characteristics of guests, i.e. 2/8 (a quarter) are counted (see Section 4.3.1.1). The only other case is for restaurant one. Since this restaurant owner while not change their quality of ingredients, all non-regular guests keep going there. This means that also the non-regular guests that have a sensitive taste keep going to restaurant one, since restaurant owner one does not change their quality of ingredients. But this is just an additional small subset of guest agents.

Lastly, Figure 4.32 shows that the regular guests do not reserve a spot. This is as expected, as they do perceive their spot to be secured. This is expected behavior from reality.

4.6.3.2 Restaurant Owner Results

The part of Expectation 3 concerning the restaurant owners is: *The restaurant owners will violate the norm by letting in more guests than allowed and/or use cheaper ingredients to increase their profit. The restaurants are profitable.* To substantiate this, the results need to show the following two patterns: 1) the restaurant is in healthy financial situation (≥ break-even), and 2) the restaurant owners operate their restaurants based on their described characteristics (using cheaper ingredients, and/or violating the norm).

1. Healthy financial situation. Overall, when looking at Figure 4.33 below we can see that all restaurants are in a healthy financial position, compared to the previous scenario where the restaurants went bankrupt (see Figure 4.21). The differences between the actual amounts of money that each restaurant is making are not important for us, as we focus on the trend and want to show that the restaurant owners chose actions to be profitable again. Consequently, we see this pattern as successfully proven.

Figure 4.33: Capital for each restaurant owner over time. The black vertical line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm response diversity scenario

2. Individual specific behavior of each restaurant owner: To address the second pattern, we look now at each restaurant owner separately, and discuss their decision making based on their individual characteristics. This means that the restaurant owners decide to violate the norm or use cheaper ingredients, depending on what is most important for them, i.e. how their characteristics relate to their needs, see Section 4.3.1.2. To do this, we first characterize each restaurant owner again, and then relate the results back to it.

In general, we will see a big drop in the beginning of the financial stability need for all restaurant owners, This is because when the norm becomes active and the number of available places is set to 50% of the total capacity of each restaurant, the financial stability need of each restaurant owner is also halved and drops to 0.5. This was a model specific choice. The graph shows a slightly higher number hidden behind the black norm activation line, as the restaurant owners still made profit that week.

We characterized restaurant owner one as someone that is very focused on the experience they provide to their guests, and for whom the relationship to their regular guests is very important. This is why the size-based restriction norm has a perceived negative impact on the security need of this restaurant owner. In Figure 4.34, we can see the drop in the security need satisfaction after the size-based restriction norm became active. Furthermore, the figure shows that the immediate reaction of the restaurant owner is to violate the norm[1] and let the regular guests more in than allowed by the norm. Additionally, the restaurant owner is not changing the quality of their ingredients, as the satisfaction level of the self-realization need (representing the quality of ingredients) remains at the highest possible satisfaction level of 1. Also, Figure 4.34 shows that the financial security need drops to 0.5 and rises to 0.6 (as the restaurant owner still makes money this week) after the introduction of the norm (hidden by the black vertical line signaling the norm activation), but crucially stays above the threshold of 0.4 and then goes up towards 1 indicating that the restaurant owner is earning enough money to be profitable. Finally, the graph shows a few dips of the compliance need. This happens when the restaurant owner is getting caught violating the norm. But since the income gained from the guests is higher than the fine for getting caught, no dips in the financial stability need can be observed.

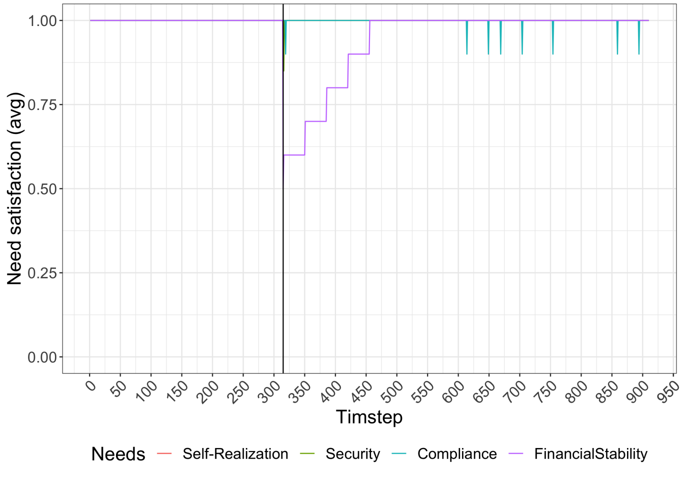


Figure 4.34: Restaurant owner one need satisfaction over time. The black vertical line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm response diversity scenario

In comparison to restaurant owner one, we characterized restaurant owner two

[1]Since the threshold for the security need of restaurant owner one is at 1, the restaurant owner wants to take action to satisfy this need.

as someone who is strongly money driven and sees the guests only as wallets rather than focusing on the experience of the guests. This means that their main focus is on the monetary impact of the norm, and thus, on the drop of the financial stability need due to the size-based restriction, as they have a high threshold for that need of 1 (see Table 4.6). The drop of the security need is fine for them, because good customer relations are not that important to them, causing a lower threshold for that need. Furthermore, with the financial stability being very important for the restaurant owner, complying to the norm is also very important, as the monetary fine of violating the norm is perceived as very harsh by the restaurant owner and thus they try to avoid it.

These reasons lead to restaurant owner two deciding to use lower quality ingredients to increase their income and not to violate the norm. Figure 4.35 shows this decision-making. The self-realization line, representing the quality of ingredients, drops to 0.66 which represents the use of cheaper ingredients. Furthermore, the security line, representing the customer relationship to one's regular guests, stays at a lowered level after the norm became active, indicating that nothing was done by the restaurant owner to counteract this drop.

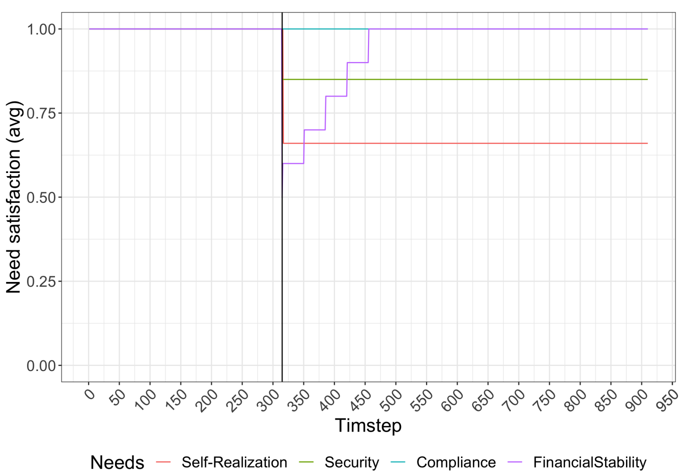


Figure 4.35: Restaurant owner two need satisfaction over time. The black vertical line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm response diversity scenario

Finally, we characterized restaurant owner three as a mix between restaurant owner one and two. Meaning that money is important to them, as well as their relationship to their regular guests. When we look now at their need satisfaction over time in Figure 4.36, we can see that restaurant owner three decides first to violate the norm and let regular guests in more than allowed to secure good customer relations with the their regular guests. Furthermore, the restaurant owner is hoping that the additional regular guests are enough to counteract the loss of income from the norm. But this is not the case and the restaurant keeps losing money, as seen by the financial security need dropping further (after the initial drop to 0.5 and replenishment to 0.6).

As a reaction to this, restaurant owner three does decide to use cheaper ingredients, as indicated by the self-realization need line dropping to the 0.66 level, to counteract the further loss of money. Finally, we can see that the restaurant owner is getting caught violating the norm, by the small downward spikes of the compliance need. This means that the restaurant owner has to pay a fine for getting caught. Paying the fine in the beginning when the financial security need drops further indicates that the restaurant owner had to use cheaper ingredients also to make it possible to potentially violate the norm again in the future, as otherwise not enough money was made.

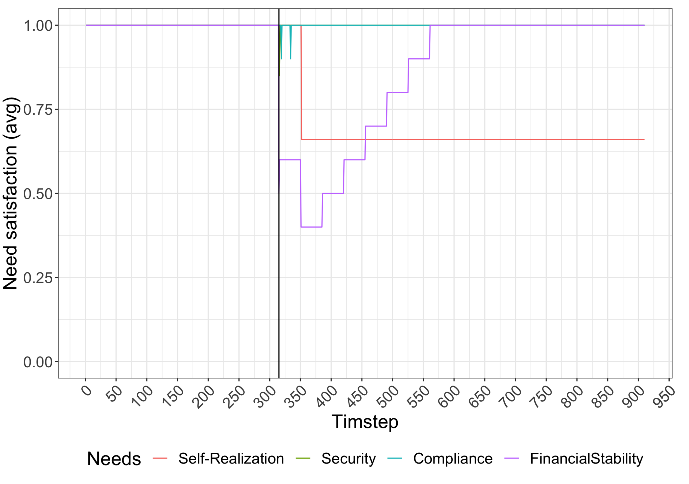


Figure 4.36: Restaurant owner three need satisfaction over time. The black vertical line shows when the norm was activated. Average of 20 repetitions with smoothed function, Norm response diversity scenario

Taking the behavior of all three restaurant owners together, we argue that the second pattern regarding the restaurant owners has been show successfully. The behavior showed that each restaurant owner is making a decision based on their individual characteristics, see Section 4.3.1.2. Restaurant owner one is focusing on preserving customer relations, and is violating the norm, restaurant owner two is adhering to the norm and is using cheaper ingredients to make more money. Restaurant owner three is doing both to secure good customer relations with their regular guests, and make more money with their restaurant.

4.6.3.3 Conclusion

In this scenario, we increased the action space of the agents to allow for diverse norm responses. This means that the agents were now able to obey the norm, obey the norm and take further actions (circumvention), and violate the norm (in the case of the restaurant owners). We addressed the following expectations (Expectation 3) in this scenario: *The restaurant owners will violate the norm by letting in more guests than allowed and/or use cheaper ingredients to increase profit. The restaurants are profitable. The guests not affected by the norm (regular guests) do not alter their behavior. The guests affected by the norm (non-regular guests) reserve a spot at the restaurant. The guests affected by ingredient change (having a sensitive taste) react to it and go to park* Since the results show that all required patterns for these expectations are met, we see these expectations as proven. A summary of the presented results in relation to our expectations for this scenario can be found in Table 4.12.

Table 4.12: Summary to check if the required patterns have been met for the norm response diversity scenario

Required Pattern	**Is the pattern met?**
Guest needs well satisfied	✓, Guest needs stable and above/around threshold, Figures 4.23 and 4.24
Guests (regular and non-regular) affected by the ingredient change (having a sensitive taste) change their behavior	✓, Regular and non-regular guests affected by the ingredient change go to the park more often after the norm became active, Figures 4.26 to 4.28
Guests (regular and non-regular) not affected by the ingredient change (not having a sensitive taste) do not change their behavior	✓, Regular and non-regular guests not affected by the ingredient change keep going to the restaurant after the norm became active, Figures 4.29 to 4.31
Non-regular guests reserve their spot at the restaurant after the norm became active	✓, Non-regular guests that go to the restaurant after the norm became active reserve their spot, Figure 4.32
Restaurants are profitable	✓, The restaurants are making money, Figure 4.33
The restaurant owners violate and/or take action against the norm	✓, Based on their characteristics, restaurant owners one and three violate the norm, and restaurant owners two and three use cheaper ingredients (norm circumvention), Figures 4.34 to 4.36

In comparison to the previous scenario, we kept the norm dimension the same, i.e. the norm became active after a third of the simulation (nine weeks), but changed the action dimension, i.e. increased the action space of the agents. Concretely, the guests were now able to reserve a spot at the restaurant, and the restaurant owners were able to use cheaper ingredients to circumvent and counteract the norm, or violate it by letting in more guests than allowed.

The results showed that the behavior in the previous scenario (Section 4.6.2) was not the preferred behavior of the agents, and a lot of potential behavior, such as people reserving spots or restaurant owners using cheaper ingredients was missed. It was not the preferred behavior, because the average needs satisfactions of the guest agents is more similar to the first scenario where the norm is not active compared to the previous scenario. Furthermore, it is obvious that the restaurant owners want to make money with their restaurant, and do not want to go broke.

4.7 Conclusion

We presented three scenarios within the restaurant model to validate our proposed PBADA agent deliberation architecture. These three scenarios allowed us to systematically exploring the different ways norms can be included - following the systematic experimentation approach suggested by Kleijnen [86] together with the trace validity approach by van den Hurk et al. [74].

For each scenario, we varied one factor, the size-based restriction norm being (de)active and the available action space of the agent. Then we formulated our expectations for behavior of the agents for each scenario. This expected behavior is based on *plausible* behavior that one would expect in reality in such situations. In other words, we made sure that the behavior of the agents using our agent deliberation architecture is matching reality. The trace validity [74] is used to make sure that the behavior of the agents in the various scenarios can solely be explained by their underlying characteristics.

In scenario one (see Section 4.6.1), we looked at the case where the norm is not active (baseline scenario). Allowing us to compare the other scenarios: to show that the introduction of each factor (the norm in scenario two and the increased action space in scenario three) did cause the observed behavior change in those scenarios. The baseline reflects a situation where restaurants were overall financially healthy and the guests went to the restaurant once or twice per week. While some people went to the park, the majority of the people preferred the restaurant over the park.

Contrasting to this, we showed in scenario two (see Section 4.6.2) that introducing a norm (restricting seating by 50%) without increasing the number of actions for the agents that allow for breaking or bending the norm leads to the restaurant going broke (i.e. not a healthy financial situation anymore). This insight is as expected, because the source of income for the restaurant are their paying guests, and reducing the amount of people that can be at a restaurant reduces the income of the restaurant. The reduced income is the not enough to cover the costs, leading to the restaurant going broke over time. For the guests, this scenario seemed to be acceptable on the first glance, as they can just go to the park to satisfy their needs of pleasure, belonging, and security, instead of going the restaurant. However, this can be unrealistic, as some people might not prefer the park, and just went there because the norm was limiting their options with the park being the only option left, as no further actions were available to the agents. In reality however, they might have preferred to go to the restaurant and do something to ensure their spot instead of going to the park.

We increased the number of actions available in scenario three (Section 4.6.3) and showed that some agents are willing to violate the norm, while others are motivated to take alternative actions to prevent the perceived negative consequences of the norm. Restaurant owner one for example perceived the norm as a thread to their customer relations and therefore decided to violate the norm to let their regular guests more in than allowed, while restaurant owner two perceived the norm as a thread to their income, and therefore used cheaper ingredients to increase the income per guest[1]. The guests that perceived a threat to having a spot at the restaurant decided to reserve a spot (again the motivation component of the norm). Norm violation was not possible for the guests, as the restaurant owner is the one responsible for adhering to the norm, and not the guests.

By meeting all expectations for each scenario, we conclude that we successfully validated our proposed PBADA agent deliberation architecture. For each scenario, we changing one factor at a time (systematic experimentation by Kleijnen [86]), and were to explain the outcomes of each scenario based on the individual behavior and characteristics of the agents (trace validity by van den Hurk et al. [74]).

4.7.1 Why the scenarios suffice to show the validity

The goal of this chapter was to verify and validate our PBADA agent deliberation architecture. Since we provide a general agent deliberation architecture suitable for many use cases, and not just for one specific situation, we should have used a variety of models to prove that our agent deliberation architecture works as intended. Furthermore, we should have implemented multiple models to identify the normative aspects underlying the decision-making of the agents. The resulting behavior in all models could then be used to recognize relevant patterns showing that the mechanisms are the same in all of the different models fitting to our proposed agent deliberation architecture.

To do this however, we need many models (to reflect a wide range of application domains), which is not feasible in the context of this PhD research. This is why we chose one concrete example, the restaurant example, to show how the mechanism of the normative aspects work and that our deliberation architecture is indeed suitable to account for them.

The restaurant example is realistic enough, because we integrate different perspectives on norms with different motivations and a different focus on the norm (guests and restaurants owners), and different actions available to them. We were able to show with the different scenarios how our agent deliberation architecture works, how the norms are processed, and where they come into play in the agent's decision-making process. Most importantly, the results of the scenarios are plausible ones, based on plausible behavior rules [7, 42, 43] (as given in our PBADA agent deliberation architecture). Plausibility is crucial for using models for developing explanations [42] which Epstein calls "*generative* explanation" [42, p.3]. In our case, this means developing explanations for the normative behavior of the agents. Furthermore, we wanted to keep the scenarios as small as possible to focus on the core elements necessary. Of course, we could have added many more different elements to the simulation, such as explicit social networks, different action options for the agents, or more perspectives. But these elements would have not changed the conclusions and outcomes of the scenario.

[1]This action is from the motivational component of the norm, as the restaurant owner is motivated to user cheaper ingredients, which is not against the size-based restriction norm.

For example, having an explicit social network would have influenced the decisions of some agents for whom their friends are important, i.e. they want to do what their friends are doing. So maybe they would still go to the restaurant to be with their friends, even though the restaurant owner did change the quality of their ingredients, and they (the guests) do not get as much pleasure anymore as before. But since their friends are going, they might go too. However, this might not always be the case, and the reverse holds true as well for some agents (i.e. the change of the quality of ingredients outweighs the decisions by the friends of the guest agent). So in the end, while the actual numbers might differ in the results, the overall outcome is still the same. The restaurant owner is using cheaper ingredients to increase their profit, and some guests (affected by the ingredient change) decide to not come to the restaurant anymore (reaction to the action of the restaurant owner).

Similarly, we could have extended the action space of the agents. Instead of just having the option to use cheaper ingredients, the restaurant owners might also be able to open their restaurant longer (or earlier) to account for the limitation of their available seats due to the size-based restriction norm. Along with this, the guests might then have the option to go to eat later or leave work earlier to go to the restaurant. While some guests might do that, and some restaurant owners would try this option instead of using cheaper ingredients, the overall point is the same. Based on the perceived negative impact on the norm, the agents decide on a course of action. The restaurant owners that perceive a negative financial impact decide on a course of action to increase their income or lower their costs. This can either be to use cheaper ingredients or extend their opening times. The guests decide on a course of action to feel secure to have a spot at the restaurant available to them. This can either be to reserve a spot or change their eating times. In the end it does not matter for our purpose to show which specific action was taken, but rather that an action was taken which counteracts the perceived negative consequences of the norm.

Adding more perspectives, such as restaurant personnel, or more needs to the existing perspectives (guests and restaurant owners), would not add any additional insights, as we argued throughout the results and showed that the behavior of the agents is based on their perceived impact of the specific behavior on their needs, and that they try to positively maximize that impact. As a consequence, adding more elements to the simulation model would have only added noise and cluttered the results. This would have distracted from the main points, and the successful verification and validation of our proposed agent deliberation architecture.

Another aspect that we did not model is the reaction of the guests to the norm violation of the restaurant owner. This would require the guest agents to have a full memory of the violations by the restaurant owner in order for them to decide whether or not they want to go to that specific restaurant. Some guests might not feel safe anymore going to that restaurant. This is a gradual process, as some guest might not feel safe after the first time, while others only feel unsafe if the restaurant owner is violating the norm quite frequently. What frequently means is also depending on the guests. Also, some guests might decide to report such violations. However, it is clear that our proposed agent deliberation architecture is suitable for handling reactions of agents to the behavior of other agents (as shown by the guest agents reacting to the change of the quality of ingredients by the restaurant owner). We showed this in the model by some guest agents getting less pleasure from going to the restaurant when the restaurant owner is using cheaper ingredients. Consequently, adding this would have only added more clutter and not changed any of the conclusions.

To summarize, we could have added many more different elements to the model to make it seem close to reality but also make it more complex. However, all models are in the end abstractions from reality [42]. This means that to decide on the degree of complexity of the model strongly depends on the purpose of the model. The purpose of our model presented in this chapter was to highlight the underlying core dynamics [42] of human decision-making in the context of norms. Since we were able to do this with our proposed agent deliberation architecture in this chapter, and the results are plausible, we argue that the complexity of our model suffices to validate our proposed PBADA agent deliberation architecture.

4.7.2 Why internal mental states are important

With the example being this simple, one could criticize now that the scenario has been kept too simple and that input-output-based agents[1] would have been enough to produce the same results. While this is a fair comment to make, we want to discuss now why this is too short-sighted and does not account for the whole power that comes from our PBADA agent deliberation architecture.

On can use less sophisticated input-output-based agents. However, the problem is that this will only work in one fixed scenario, i.e. the one that the rules (the input-output connections) were made fore. This means that they can only be used in a very limited and fixed environment. This is where complex architectures, such as our proposed agent deliberation architecture come into the picture. They can be used in complex and changing environments, and various scenarios [65, 83, 116, 140].

With motivators as the foundation of behavior, we allow our agent deliberation architecture to be used in various environments, as the behavior of the agent is based on them. Furthermore, we allow for norms to be modifiable objects in the simulations. While we kept it fixed in our model experiments, one can easily see that if we would add another norm, our agent deliberation architecture does work fine without any additional effort that has to be made. This is in strong contrast to rule-based input-output-based agents. For those agents, we would have to go through the whole decision tree and see where changes would have to be made, if new rules would be added. Furthermore, any rule conflicts would need to be resolved by hand.

Another important reason is that (legal) norms do not exist in a vacuum. They are part of a complex web [124] of and interact with various other social rules and concepts, such as other (social) norms, values, habits, goals, social practices, social expectations (see Section 2.2 in Chapter 2). This means that when having simple input-output-based agents, those rules must account for those social concepts as well and define a behavior for that. For example. if the user wants to add another norm to the simulation, it needs to be well defined how that new norm interacts with other

[1]We mean here agents that are just reactive based on the input they get without further deliberation about it, i.e. simple if-then-else statements. One input directly produces one output.

potentially conflicting social rules, such as habits. If a norm is conflicting with a habit, it needs to be defined if the habit is followed and the norm violated or vice-versa. The same argument holds for conflicting social norms. Policies (legal norms) can conflict with existing social norms [65], and therefore it needs to be defined for every possible situation (no matter how big the chance of the occurrence of that situation is, as otherwise the simulation could crash) what action the agent is choosing and which norm takes prevalence over the other. If no action is defined for every situation, the simulation will break if that situation occurs.

The previous two paragraphs show that input-output-based agents make it difficult for non-expert users to use the simulation in a for them meaningful way and modify the simulation. This would require that the non-expert users would be able to change the code of the simulation which is both not suitable and not possible, as they do not have the required knowledge and capabilities to do so. Since our PBADA agent deliberation architecture allows for modifiable norms, we discuss the support required for the non-expert users in detail in the next chapter. We will show there that a very high-level of support is necessary for them to use and alter the simulation.

Chapter 5

User Interaction Tool

5.1 Introduction

To support policymakers in their decision making it is not only crucial to have realistic - human like - agents but also to understand what is going on in the simulation and *why* it is happening, i.e. why agents behave in the way they do and why they respond to norms (policies) in the way they do[1]. Only then can policymakers fully trust the model and use it in their decision-making process. While part of the trust comes from the successful verification and validation of a model (in our case our proposed PBADA agent deliberation architecture as discussed in the previous chapter), trust also comes from the interaction of the user with the simulation. This requires a high level of usability of the simulation, as discussed in the second challenge of the introduction of this thesis (see Section 1.2 in Chapter 1).

Improving the usability is however not a simple task of adding some buttons, and graphs in the interface on top of the agent architecture used in the model. Rather, the challenge to improve the usability comes from integrating norms into the agent's decision making process. Our proposed agent deliberation architecture shows that norms come into play in multiple stages in the agent's decision-making process. This needs to be made accessible for the user of the simulation, so they can understand why the agents are making the decisions they make. This means that one challenge for increasing the usability of the simulation is a methodological one, showing how norms are integrated in the agent's decision-making process, and how they affect this process. Participatory approaches, such as companion modeling [44], are not suitable here, as their focus is on understanding the groups involved. The norms are not modeled, and discussed explicitly in these approaches. They are just seen as rules of the game played during the modeling process which makes it difficult to see the potential consequences a norm might have.

The other challenge is a technical one in terms of how the agent deliberation

[1]The research presented in this chapter is based on and partially published in our following papers: [79, 82]

architecture is implemented. For the user to modify the simulation and use it in a form them meaningful way, they need to be able to adapt the simulation and add new norms or modify existing ones [82]. However, policymakers are non-expert users, and not versed in coding and thus not able to modify the simulation on a code level to, for example, add the code for a new norm. This is where we see the place of the modeler. The modelers are the ones that take the requests from the user and implement them in the model. This means that usability also means to improve the communication between modelers and non-expert users.

To tackle this multi-dimensional challenge, we propose the need for novel interaction tools and a new research agenda. This entails not only the focus on the non-expert users of the simulation, in our case policymakers, but also on the modeler of the simulation. Novel interaction tools need to empower the user to use the simulation in a, for them, meaningful way (by allowing them to modify and analyze the simulation), and act as a communication tool between the non-expert user and the modeler to allow for a fruitful, and efficient collaboration and to resolve potential conflicts that the newly desired norms (from the policymakers) can bring about.

In terms of the research questions for this thesis, this chapter answers the following ones. **RQ 3: How can users be supported to change norms during the simulation run?** and **RQ 4: What support do non-expert users need from a user interaction tool to be empowered to use the simulation in a for them meaningful way?**

To tackle this very difficult task, we will layout the difficulties in the next section, followed by providing a path towards an interaction tool. Along this path, we will outline potential solutions, and conclude by motivating a new research agenda to focus on such interaction tools.

5.2 Why is it difficult to support non-expert users and modelers?

Supporting policymakers and other non-expert users is not a trivial task of simply building an interface on top of the model. The purpose of the interaction tool is to support modelers in the interaction with policymakers, a both of them are required for modifying the simulation (the policymaker for the desired modifications and the modeler for implementing those desired specifications). To fulfill this purpose, the interaction tool needs to provide a common ground for discussion, and create a shared mental model of the simulation. This leads to the two main challenges that make it so difficult to support policymakers (and other non-expert users) and modelers: bringing the two worlds, the policymaker world and the modeler world, together, and provide a communication tool to support the modification of norms in light of the strong connection between norms and the agent's decision making process.

5.2.1 Two disconnected worlds

Figure 5.1 shows the disconnect between the world of policymakers, and the world of the social simulation modelers and the social simulation model, as an expert system, in the middle as a potential bridge between these two worlds. One of the reasons for the disconnect between the world of the policymakers and the world of the modelers is the plethora of background knowledge that the policymakers lack compared to the modelers (with respect to social simulations) [123] potentially hindering them to get quick insights into the agent's behavior. In particular in our domain, where we (the modelers) design social simulations that allow for agents to deal with (new) norms in different ways, i.e. the agents can comply, violate and work around a norm, we use a lot of concepts, such as norms, goals, actions, planning, that need to be understood.

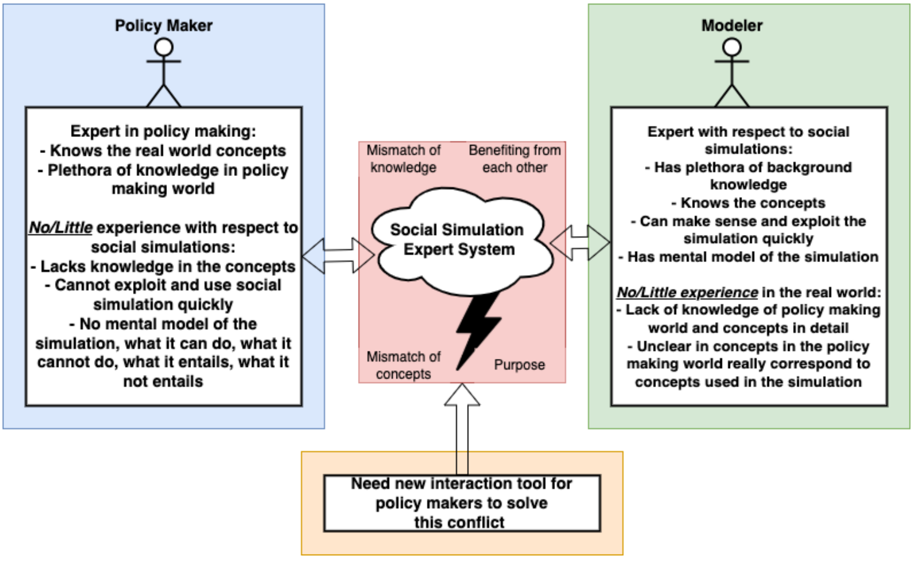


Figure 5.1: Social simulation as an expert system in the middle and potential points of friction in the corners highlighting the gap between modelers and policymakers, adapted from our paper in [79]

In contrast to that, policymakers are experts in policymaking and mostly having no/little expertise in modeling. Meaning that they have a clear and crisp understanding of what policies are, and how they are to be differentiated from other social rules (such as conventions). Furthermore, policymakers know the real world concepts and have a plethora of knowledge in the policymaking world. But they lack the knowledge required in the modeling world. Modelers on the other hand are experts in modeling and not in policy making. We (the modelers) have some idea on how to transfer real world policies into norms (usable in social simulation models) using conceptualizations of norms but there is no guarantee that this is actually matching the reality.

Consequently, we (modelers) need to be aware of the differences in background knowledge and the language used. As modelers we may be well versed in using concepts such as goals, actions, planning and norms which is too easy to be taken for granted and unconsciously used every time we analyze or interpret a problem or result [123]. As a result we need communication tools for policymakers that empower them to use the simulation in a meaningful way for them rather than having to submit to its (the simulations) complexity [123].

The key challenge for supporting policymakers is to enable them to get a mental model of the simulation, meaning to have an understanding what the simulation can do, what it contains, how the agents reasons, and also importantly what is not in the model. This means that it is very important to understand for the user that social simulations can only provide a 'reasonable' abstraction of reality. This abstraction and what is 'reasonable' strongly depends on the purpose that the simulation is made for, for example disease transmission in COVID-19 (see here Dignum [36]).

Having this understanding, that the model is a *reasonably* simplified representation of reality, is crucial, because it helps to understand that for the given *purpose* of the model some parts of the model are fixed (to ensure its functioning) or simplified, while others must be complex. This can vary from model to model however. For example, for our purpose of supporting policymakers, we require human-like realistic agents and thus a high degree of complexity of the agent's deliberation, as our proposed PBADA agent deliberation architecture in this thesis showed. Other models, such as land use [60], or marketing [113], suffice with agents that have a comparatively simpler decision-making process of the agent.

While realizing and bridging this gap is important, it is not the only challenge that needs to solved with a potential interaction tool. The agent deliberation architecture which we developed in this thesis, shows that norms are tightly connected to the agent's deliberation process and come into play at multiple stages. To support the user in this understanding, we show in more detail how norms and perspectives are connected in the next section.

5.2.2 Bridging the gap between modelers and policymakers

The challenges outlined above reinforce the need of an interaction tool to be the central point for the policymaker to interact with the simulation and modify it, and also communicate with the modeler, see Figure 5.2. Therefore, it has to enable the things that the policymaker needs in terms of the interface and model. This means that the policymakers must be able to translate their inquiries and demands into the world of the model via the interaction tool. To tackle this challenge, we start with discussing what the policymaker wants to see and then translate their potential questions into the social simulation world (i.e what it means in the model). Once we are in this world, we use these questions to discuss what the policymaker can see and thereby moving back out of the social simulation world.

Furthermore, Figure 5.2 shows the connection with the modeler, as the interaction tool is also the communication point here. This means that the interaction tool needs to provide the necessary information from the user (policymaker) in such a way that the modeler can use it to modify the simulation, as the modeler is still the one that has to modify the code basis of the model if for example a new norm has to be added. To make this process as smooth as possible, the interaction tool needs to enable discussions about potential conflicting situations that can occur and have to be solved.

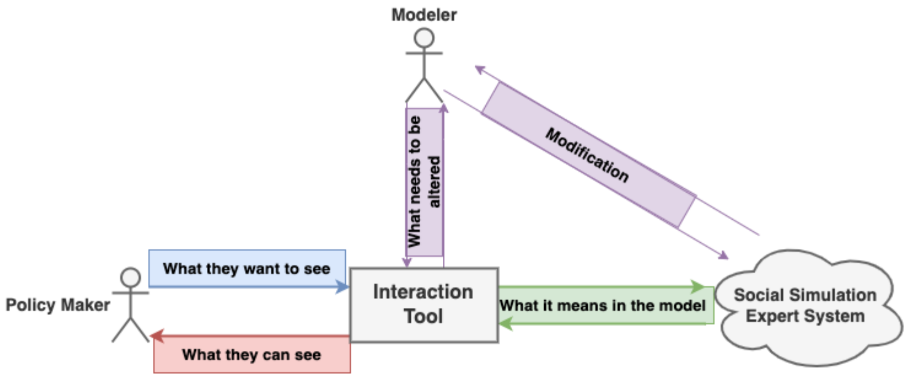


Figure 5.2: Interaction and information flow with the interaction tool in the center as the mediator between the policymaker and the simulation, based on our paper in [79]

5.2.2.1 What the policymaker wants to see

This is the point where the policymaker starts to interact with the simulation (blue arrow in Figure 5.2) through the interaction tool and tries to seek answers to their questions which in our norm focused approach could be:

How does the new/modified norm conflict with existing norms?

Norms do have a fulfillment and a violation condition [78]. If the fulfillment condition is met by the agent they successfully complied to the norm and if the violation condition is met, the agent violated the norm. Two norms are in conflict if the fulfillment condition of one norm can only be met by simultaneously also meeting the violation condition of another norm, given that they are active in the same context. The context is very important, because the conditions of two norms can be in conflict but if they are not active in the same context, it does not matter. For example, being at the office at 8AM while simultaneously having to be at home at 8AM is in conflict. But having to be in office at 8AM and having to be at home at 8PM is not in conflict. Furthermore, it can be possible that the norms in themselves are not conflicting with each other (which can even happen with just one norm), but situations can occur where they conflict. For example, having to be at office at 8AM and obeying the speed limit are not conflicting, unless the agent is late and must now make a decision between being at the office at 8AM and violating the speed limit or the other way around.

Can the policymaker assess how the behavior (of agents) changes?

A policymaker adds a new norm or modifies an existing one with the intention to influence the agent's behavior. Similar to the real world where norms are used to guide and determine the people's behavior. Therefore, the policymaker is interested to see if the norm change has its intended effect on the agent's behavior. To do so, they need to see not only beforehand how the behavior of the agents might change but also after the implementation of the norm.

Can the policymaker asses why the (new) norm is violated?

Similarly to the two previous questions, it is very important for the policymaker to understand how likely it is that a (new) norm is violated and why that is the case. There is no simple answer to that. Maybe the new norm interferes with existing norms both legal or social that are more important to the agent. Furthermore, it might be possible that the actions forbidden by the new norm are more important to the agent than the norm. Also, it might be possible that the violations occur regularly or are just exceptions and the norm is usually followed. The reasons can be various and the simulation must provide insights.

5.2.2.2 What it means in the model

An interaction tool must now map these questions onto the concepts used in the model and then answer back (green arrows in Figure 5.2). With respect to norms, this means to use deontic logic: Obligations (must norms), Prohibitions (must not norms), and Permissions (can norms) [34].

To show how to connect these types (of norms) to the questions from the previous section, we use the second question, assessing the behavior change of the agent, as an example (the other questions work analogues). Given a new or modified norm, it can be possible now that the desired action of an agent became forbidden (is now prohibited) or a different action became obliged. If the originally desired action is more important compared to adhering to the new/modified norm nothing changes, but if the norm is important, the behavior of the agents changes in a certain way. For example, the agent can chose an action that is complying with the norm and fulfilling the same purpose as the previous action or be motivated to change their behavior in other ways to circumvent the perceived negative consequences of the norm. It is important to note here that if a norm makes an action that the agent does anyway obliged, nothing changes. Especially this case, when the behavior of the agent is not changing, is very difficult to show, as it can be unclear for the user of the simulation to see if the agent is taking the new/modified norm (policy) into account at all in their decision-making process, or if the new/modified norm did not have the intended effect, and nothing changed. To show why the agent is potentially not changing their behavior, how and in what way a norm has been taken into account in the agent's deliberation process, and how the potential behavior changes look is the responsibility of the red arrow (Figure 5.2), discussed in the next section.

5.2.2.3 What the policymaker can see

In the last step (red arrow in Figure 5.2), the interaction tool has to translate the data gathered from the model back to the real world, and into the terminology of the policymaker. Furthermore, based on how the answers match with the initial questions of the policymaker, the interaction tool needs to provide ways to interact with the simulation such that the answers (red arrow) match the questions (blue arrow, Section 5.2.2.1). For example, to see the likelihood that a certain norm is violated, we have to identify if the policymaker wants a simple ranking of norms based on their violations or if they want to see the violations over time, or just an abstract analysis showing potential norm conflicts and conflicting situations. Furthermore, we have to identify the support that policymakers need to modify an existing norm or add a new norm. It might be possible that they want to see at every step in the formalization process how and if the current formalization is conflicting with an existing norm or which agents are potentially affected by it. Importantly, it might be possible that the policymaker wants a preview within the overall simulation, with only a few agents and norms to verify if the current formulation works as intended or to see potential norm conflicts and conflicting situations in the existing sets of norms without having to run the whole simulation, and thereby being able to focus only on a small subset of the simulation. This leads to the following list of very concrete questions that need to be considered: Does the policymaker need: Textual support? Graphical support? Previews? A novel agent inspection tool? Other ways?

5.2.2.4 Modification of the simulation

A crucial part of this process is also the communication with the modeler (purple arrows in Figure 5.2). The modeler is responsible for making the changes in the code if necessary (highlighted by the purple modification arrow in Figure 5.2). For example, when adding a new norm, the modeler needs to write the code for it. To do this and in case the user (policymaker) wants information that is not covered to so far by the model (e.g. a new reporting variable), the interaction tool needs to provide a common language and specification that can be used by the modeler. Consequently the interaction tool plays a key role in managing the expectations, providing a clear conceptualization of what the user (policymaker) wants from the modeler, where potentially conflicting situations occur, due to the desired (normative) changes, and the decisions that have to be made there, and to resolve any ambiguities in the expectations, language, and mental model used by the both, the user and the modeler.

To give an example from our own experiences in our ASSOCC [37] project. The communication between the project members that did run the scenarios, for example closing the schools [73], or investigating the potential impact of policies on consumer economics and the supply chain the economic impacts [98], and the project members working on the simulation model itself, could only be so clear, efficient, and effective, because the people running the scenarios where present in the project meetings and thus knew what the model can do and how the model works. This means that they had the *same mental model*, as the ones working on the model. However, this is not feasible in reality. The policymakers cannot attend every modeling meeting. Consequently, this task of providing a shared mental model falls to the interaction tool.

5.2.3 Connecting Norms and Perspectives on Norms

Having discussed now and in mind potential questions that policymakers can have, we also need to look at the PBADA agent deliberation architecture to see how it can help to provide answers to these questions. In a way, we now look at Figure 5.2 from the simulation point of view. To do so, we want to reflect on the deliberation architecture, and discuss how the norms connect to the different perspectives from the angle and with the purpose of understanding what we can use from the architecture, and how we can use it to support the non-expert user in understanding the normative behavior exhibited by the agents. This is also to address the arrow in Figure 1.1, in the introduction of this thesis, which points from the architecture to the potential interaction tool, as the interaction tool is influenced by the agent deliberation architecture.

To do so, we can look at Figure 5.3 which shows two norms, and two perspectives (guests and restaurant owners, based on the previous chapter) The restaurant owner perspective is abstracted away (with only the financial stability need remaining) to increase the readability (as it would otherwise be too cluttered), and most importantly to really allow us to focus on the guest perspective.

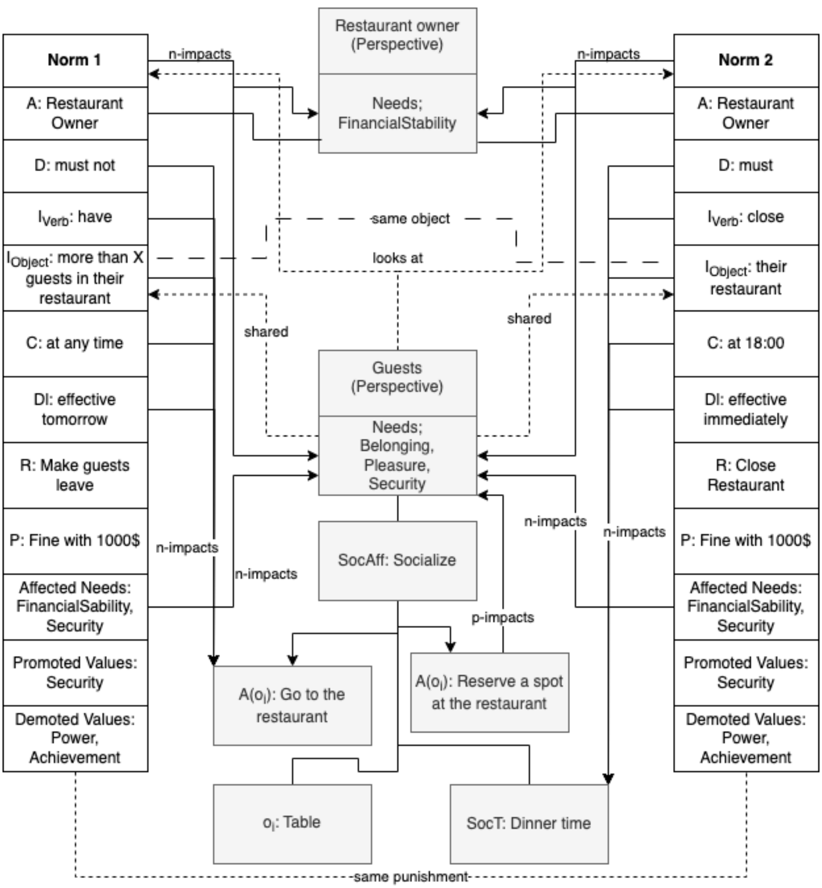


Figure 5.3: Visual representation of norms and their connection to the perspective on norms, adapted from our paper [82]

This figure shows that the restaurant owner is responsible to adhere to both norms (A). Furthermore, norms that share the same object (I_{Object}), and prominent punishments (here 1000$ fine) can be identified (same, dotted line). The graph also shows that the norm to have less guests in the restaurant negatively impacts (n-impacts, solid arrow) the financial stability need of the restaurant owner, as fewer guests can be present at the same time. The guests share the restaurant (I_{Object}) with the restaurant owner (shared, dotted line). Based on this shared relationship, guests can now look at the respective norm and all the relevant elements for them to see how that can potentially impact their behavior (looks at relationship, dotted arrow). They (guests) now see that their social affordances (the social action, and purpose of the physical actions)), in this case to socialize, are negatively impacted (n-impacts, solid arrow). Norm one (the size-based restriction norm) results in fewer tables available and thus a negative impact on going to the restaurant, and norm two (the mandatory closing time norm) potentially alters the dinner time, due to the mandatory restaurant closure. Consequently, the figure shows that the two norms negatively impact the pleasure and belonging needs of the guests to have dinner at the restaurant and socialize with their friends and family (n-impacts, solid arrow) respectively. This information can then be used by the guest agent to find an action that has also the purpose (social action) to socialize, and is impacting the negatively impacted needs by the norms positively. In the figure this is exemplified by the action to reserve a spot at the table (p-impacts, solid arrow).

This discussion shows that the connection and interaction between norms and perspectives is very complex. A potential interaction tool needs to make this complex connection clear, and intuitive to understand to provide valuable insights on the normative behavior of the agents for the non-expert user. This is also to use our PBADA agent deliberation architecture as a supporting tool to answer the questions raised in the previous section.

5.3 Towards an Interaction Tool

To bridge this gap between the *non-expert users* (policymakers), and the *expert system* (the social simulation model), we need to analyze the potential requirements from two ways. The first one is looking at the existing work that has already been done[1], meaning to analyze the current state of the art in the field of social simulations (see the next section, Section 5.3.1). However, this alone is not enough, as we unavoidably will fall in the same trap that we discussed above, namely that we will only analyze the potential requirements from our point of view, and use our background knowledge (albeit subconsciously) to assess the existing literature. To solve this problem, we will additionally perform an empirical analysis via a focus group study to gather potential requirements (see the section followed by the theoretical analysis, Section 5.3.2), and compare it with the requirements that we analyzed to get a more complete picture of what is required by non-expert users.

5.3.1 Theoretical Analysis - Literature and Existing Work

A notable earlier work we want to mention is the work by Broekens et al. [14]. They focused on self-explaining BDI agents in their work and which kind of explanations

[1]We touched upon this in Section 2.5 in Chapter 2

are most natural for users. The authors pointed out in their work that humans explain their behavior using folk psychology with ones underlying mental states [14, 85]. Furthermore, the authors also state that humans use reasons to explain intentional behavior [14, 93].

While the authors focus on BDI agents and the resulting goal tree, connecting the goals (desires) believes and actions together, the general approach that "[...] the mental concepts responsible for an agent's action are also used to explain that action" [14, p.30] can be very useful. For example, when the user wants to know why the agent chose a specific action, the agent could say (based on the restaurant example from the previous chapter): *we chose the action to go to the restaurant and reserve our spot there because our belonging, pleasure and security needs needs are salient, and to guarantee a spot we reserved one*. Of course, it has to be seen how this explanation can be as natural as possible.

This form of explanation goes in line with the work done by Harbers et al. [64] who also worked on self-explaining BDI agents. By traversing the goal-believe-action tree bottom up their explanations from the agent's point of view can be summarized as: I (agent) $< action >$, because I $< goal >$ and I $< believe >$. Note there that this is similar to what we proposed with the needs and norms as underlying reasons for the actions (without a tree-like structure). The authors [64] proposed various heuristics to the produced explanation, such as "[...]presumed known information is left out [...]" [64, p.6] or resolving ambiguities by going up the goal plan tree until the lines fo the actions meet, and then giving a reason why one plan was chosen over the other [64]. While these can be helpful and we agree with the general statement by the authors that we (the modelers) need to be careful to not overload the user with information when explaining the agent's behavior, we (the modelers) also need to be careful which heuristics to use. For example leaving out presumed known information can be dangerous as for some users the information might the known while for authors it is crucial that the agent includes that part in their explanation.

The GAMA platform [147, 148] is also an important work to mention. The GAMA platform provides an open-source environment for developing spacial agent-based models. It provides modifiability of the agents [148], as well as various visualizations that the user can chose from for the simulation run [59, 148], such as different kinds of maps. There are several problems however. The first one is that the modification support for non-expert users is not well developed. Users can modify agents, but the way they have to do it, is still very technical. For example, the dialog for adding new agents presented in [148] requires from the users to define the relevant variables including the type and initial values. While this is required for a working code, the dialog crucially does not provide any support on what would make sense here. For example if the user wanted to add a variable called 'age' to the agent, the interface does not provide support in saying what type the variable age should be or the initial value it should have, meaning that the user has to know this by themselves. But the user *does not* have this knowledge and needs to be supported in this process. The second issue is that there is no special focus on norms. The focus is only on the agents, and not on the norms. This means that the norms are not easily modifiable but rather only exist in the code, as there is no way to model norms to later be added in the simulation.

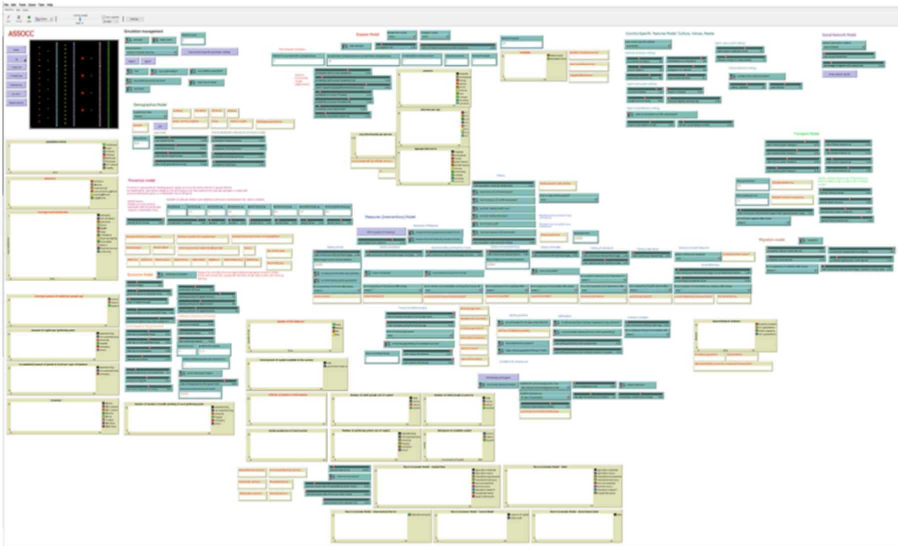


Figure 5.4: NetLogo interface as composite image of several individual images, taken from the ASSOCC book [123]

Finally, the GAMA platform only works for spatial agent-based models. However, not every model and social simulation needs to have spatially explicit agents, as shown by our ASSOCC project [37]. This severely limits the applicability and potential use cases for the GAMA platform [147, 148]. One last thing that we want to mention is that the authors in [148] mention that in future work they want to connect the GAMA platform to a game engine like Unity[1] to further provide high-quality 3D displays, and interaction.

Another important work that we want to mention here is the ASSOCC project [37, 123]. The reason that we mentioned this project here is that this project did not only focus on improving the usability and communication with non-expert users, but also showed and highlighted the advantages of using a game engine like Unity[2] for supporting non-expert users. A Unity interface [123] was built, as the NetLogo [162] interface had become so complex, and consequently, hard to use, especially for non-expert users, see Figure 5.4. The figure shows that the interface is highly complex and overwhelming, even the ASSOCC project participants, the expert users, were overwhelmed and could hardly use it in the end. Furthermore, the figure is a composite figure of several images, as the whole interface at once could not fit on a single screen anymore. In addition, every green element in the interface represents and parameter that can be modified.

The key requirements for guiding the development of the Unity interface were as follows [123]. The first group of requirements were focusing on empowering the

[1]https://unity.com/

[2]https://unity.com/

user and not overwhelming them. Non-expert users need to feel empowered to use the simulation to get meaningful insights and not submit to its complexity. This also means that the interface needs to hide the complexity as much as possible to not overwhelm the user. Examples for empowering the user, and hiding the complexity can be seen in Figures 5.5 to 5.7 which show different graphical elements of the Unity interface. Figure 5.5 shows the bottom part of the interface with buttons to highlight different graphs at runtime (for the figure all of them are displayed). This does not only meet the aforementioned requirements but also, visualizations that are expected are shown, such as line graphs for infections over time, to keep it easy. This is another key element when designing an interface, as the interaction must feel natural to lower the cognitive burden and should not contain unexpected surprises.

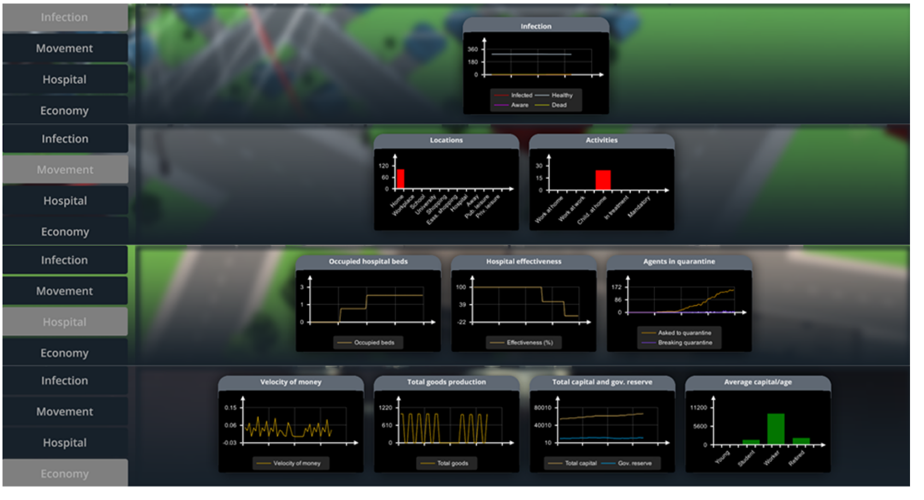
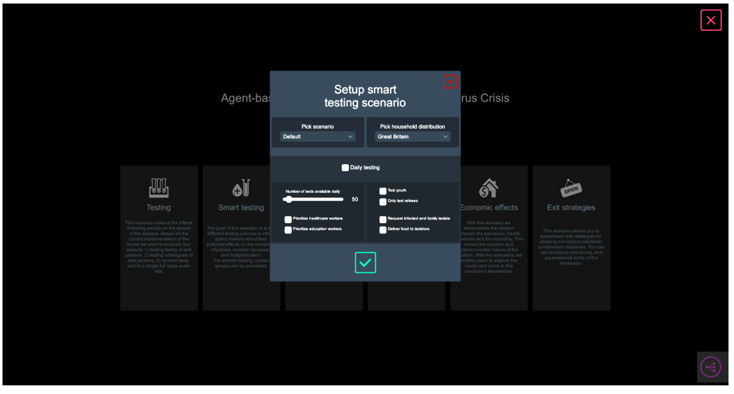


Figure 5.5: Various strips of data can be selected during the run of the simulation while the others are not visible, taken from the ASSOCC book [123]

Figure 5.6a is very interesting for these requirements as well. The figure shows the different screens for starting the scenarios. The first one is to select a scenario with a short description of what the scenario means and the second one (Figure 5.6b) to adjust the specific parameters of the selected scenario. This does not only adhere to the aforementioned requirements of not overwhelming the user, empowering the user, and hiding complexity, as otherwise the information would have been displayed all at once, which would have been very overwhelming for the user. This hiding and piecewise showing of information is a very powerful and important mechanism that we will use later in our preliminary work on an interaction tool as well.

A very important requirement is that of being realistic and making the limitations clear. As we discussed throughout this thesis, social simulation models are only an *abstraction* of reality. Therefore, the model must make it clear what is included in the model and what not. This argument of making the limitations clear was also mentioned by Melchior et al. [99]. In the ASSOCC model, this was realized by a map floating

(a) Scenario cards, taken from the ASSOCC book [123]



(b) Parameter settings, taken from the ASSOCC book [123]

Figure 5.6: Start Screen selections, taken from theASSOCC book [123]

in space, see Figure 5.7. The city is floating in space with the agents being visualized by flight-lines. This visualization projects the realism that a city is simulated, but also makes the limitation clear that no spaciality is explicitly modeled, as the agents are represented by flight-lines, suggesting that they are teleporting between the locations, and not travel on the streets in the map.

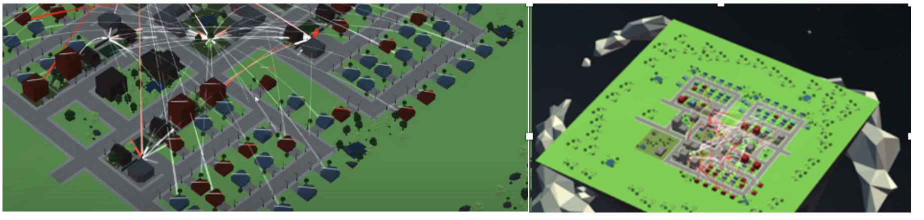


Figure 5.7: Unity map in space as composite picture of the whole map on the right (taken from ASSOCC book [123]), and showing the flight-lines on the left

Finally, to allow for the modifiability of the simulation, we want to mention the work of Păstrăv [124]. Păstrăv's work made a significant step forward in allowing the simulation to be modified. They focused on a strongly modular approach. With this approach, each element can be addressed in the editor by itself, because the interdependence between the different elements in their architecture is reduced. Their interface and model is also built in the Unity[1] game engine, and the Odin Inspector[2] for the creation of custom interfaces of the Unity editor. The interface of the Unity editor provides a textual way to specify the various elements and flip through them. Furthermore, Păstrăv argues that the model builder and inspector can be used to bridge the communication gap between modelers and non-expert users, as "further narrows the gap between the design and implementation phases of model design" [124, pp.97].

Păstrăv's approach has two major drawbacks. The first one is also mentioned by them and provides a general problem: The additional model code required [124]. This problem arises when the user wants to add an element, e.g. a norm that is not yet existing in the simulation. Then the modeler needs to write the code for that. This is why one purpose of our proposed interaction tool is the improved communication between the (non-expert) user and the modeler, such that changes can be made efficiently and to the extend that the user intended them.

The second major drawback is the lack of support for the user. This means that the interface provided is only a textual interface. The user can click and select elements, but cannot get further information what this element means or entails. This essentially means that the user can only adapt the simulation without being able to understand what they are adapting. This is a major drawback, as non-expert users are not sure what exactly they are modifying then and just go by feeling what they think it means. They are basing their decisions on the assumptions what it means for them. But in reality it can mean something completely different in the simulation. So this is something that

[1]https://unity.com/

[2]https://odininspector.com/

we need to approve upon, as explanations for the user are necessary. Another major drawback is that this textual interface is only in the Unity[1] editor. This means that in order for policymakers to modify norms, they need to have a basic understanding of the editor. This is highly problematic, as those editors are usually developed with expert users in mind and their specific needs. This also means that modeling experts who are new to the Unity game engine also might have problems in the beginning using it, even though they have the theoretical background knowledge about norms. Nonetheless, the work by Păstrăv [124] serves as a great starting point for understanding how the functionality in Unity can be leveraged to connect user inputs to the elements in the simulation.

5.3.2 Empirical Analysis - Focus Group Study

To complement our requirements analysis, and to understand what the non-expert users need from their perspective, we performed a focus group study to gain insights into the potential requirements of an interaction tool with respect to norms[2]. The focus group study also serves the purpose to avoid the pitfall to analyze the potential requirements only from our modeling expert point of view, thereby using our background knowledge, and not taking the non-expert user view into account.

5.3.2.1 Method & Preparation

To gain insights into the potential requirements of a norm focused interaction tool for policymakers, we choose to do focus groups. Focus groups are "a way of collecting qualitative data, which - essentially - involves engaging a small number of people in an informal group discussion (or discussions), 'focused' around a particular topic or set of issues" [163, p.177]. One major advantage of this informal setting is that it allows the participants to feel safe which can result in them being more open in the interactions (which can provide important data), creating spontaneous responses, and applying experience from personal problems to provide solutions [17, 41, 112, 119, 153].

For the focus group study in itself, we used the scenario analysis approach based on Ramanath & Gilbert [126]. To do so, we constructed a scenario that the participants went through in a role playing approach. In the scenario the participants where playing the role of a mayor, i.e. a policymaker, of an average European town. The town is governed by laws (legal norms) as well as people upholding social norms between them. These norms in the scenario were simple norms. For example, *being in time for a meeting* or *socialize with colleagues over a beer on Fridays* would be social norms; whereas *adhering to the speed limit*, or *no drunk driving* are examples of the type of legal norms.

The overall narrative for the participants was that the town had a problem with drunk driving which they, as the mayor, promised to tackle. To do so, they were provided with an imaginary social simulation platform to support them in their decision-making

[1]https://unity.com/

[2]The results of this focus group study are also published in our paper in [79]

process. It is important to note here that the whole study was a *conceptual* one, meaning that it was done on paper with no existing simulation platform. The goal was to already identify requirements before designing such a simulation platform. The scenario and the tasks can be found in Appendix A of this thesis.

Procedure: After setting the stage, the participants were provided with different tasks to discuss in the group. In total, we had six questions that were split up into two main themes which we distributed beforehand between the groups. Each theme contained three questions:

Theme 1 Gathering information and adding a new norm: After presenting the hypothetical simulation and the problem of drunk driving, the theme contained questions regarding what information and in which way certain information would be helpful to assess the problem. Additionally, the focus group participants dealt with the way the system can support the user to add a new norm to tackle the problem; but also how to find out why potentially unexpected consequences occurs after the introduction of the new norm.

Theme 2 Identifying and solving norm conflicts: Groups dealing with this theme were presented with a concrete norm conflict. Questions the focus group participants dealt with concerned how this specific conflict should be presented (i.e. to show that there is a conflict in the given situation), how the simulation can support the user in solving conflicts, and how overall norm conflicts (any conflict) and violations should be presented to the user.

We ran six focus groups (17 participants) of which half had theme one and half theme two. Each group consisted of two to four participants. Each focus group took 60 minutes, of which about 15 minutes were reserved for each of the three questions and the remaining time was used for the introduction and debriefing. We designed this focus group study in-line with the recommendations of focus group design: a) to have a focus group session last between one to two hours, and b) to have a small amount of participants per group to make use of the expert knowledge existing within the group [119].

As participants, we invited researchers from different research disciplines: law, computing science, education, philosophy, psychology, political science, and feminist philosophy, to ensure a variety of backgrounds to get as many and various inputs as possible and not having the risk to have only input from one viewpoint. Since we chose to do this exercise with researchers of diverse disciplines, we thus did not engage with policymakers yet as they often have busy schedules and the exercises may be considered too abstract with no concretely implemented application for their local use.

The analysis method was as follows. We used the notes from the participants together with the recording to identify all arguments made. For each argument we decided to include or exclude it based on not only the amount of agreement it got within the group but also the momentum behind each argument. This ensured that also the arguments of more silent and seemingly shy people where included.

5.3.2.2 Results

Given that we want to provide overall requirements for a norm focused interaction tool, we are giving the general results and are not dividing them by question here. Furthermore, the answers overlap between the individual questions and the same arguments have been used for the different questions, for example a common argumentation pattern by all participants was “as we said for question [...] we want [...] here as well”. The complete list of results can be seen Table 5.1:

Table 5.1: List of results of the focus group study, in no particular order.

Color Coding	Spotlight function	Only show why on demand
Graph maker tool	History of changes	Flow-graph of agent’s decision making
Color change in graph	Warnings of potential cascading effects	Sims inspired graphical information
Grouping of norm conflicts	onClick on the agent	ReadMe/manual at the beginning
Narrative based on single agents	“Profiler”	Disclaimers/hover functions
Filters for agents	Warnings about norm intrusion	Information per neighborhood
Behavior space plus	Layering of maps	Recording/Re-Run with special focus
(Aggregated) Log	Heat maps	Dynamic norm graph

We can see from the list of results that the focus overall was on the exploring the simulation. What we mean by this is that the participants mentioned many aspects, such as color coding or filters for agents or hover functions to see the definitions or what the colors represent. Furthermore, they wanted to see various kinds of maps, such as traffic flow maps or heat maps and potentially layering them. Also, a tool that combines the NetLogo Behavior Space[1] together with the statistical analysis afterwards in one tool was discussed.

While many suggestions were made related to graphs, we considered another suggestions particularly relevant: the onClick agent inspection function. Some participants argued for understanding how agents solve norm conflicts and also whether or not to violate a given norm in favor of an action to use a flow-chart or decision-tree-like structure that is shown when clicking on an agent. This was considered to support the user of an interaction tool to get insights into why the agent made their decision in that way and thus provide potential answers to the *real why* of why things are happening the way they do in the simulation. Further information that could be available when clicking on the agent, as mentioned by the participants, could be their current goal, their current needs satisfaction, their next action, and so forth.

[1]https://ccl.northwestern.edu/netlogo/docs/behaviorspace.html

Nonetheless, we want to shortly mention some results regarding graphs, as they show that we were in our ASSOCC project [37] on the right track: 1) a 'graph maker' tool, where users either can select the data they want to see and then based on the selected data the simulation shows possible graphs that can be created or vice versa, and 2) the option to have the color changed in a graph based on the activation status of a norm. The graph maker provides great interactivity with the simulation as well as freedom for the user to see the generated data in many different lights. The main reason mentioned by the participants was that this would create more insights for them and aid them in understanding why things are happening in the simulation. The color change in the norm graph idea mentioned by some of the participants was that in order to contrast agent behavior before and after a certain norm got (de)activated, the color of, for example, a line graph could change. Meaning that if there is for example a graph that shows the amount of people at a specific location over time, the line could be blue, and once a norm targeting that location gets (de)activated, the color of that line changes.

Another interesting suggestion concerned a dynamic norm graph by some participants. With this the participants meant a graph where the nodes represent the norms and the edges connect norms that share the same object. In such a graph, norms that are connected by edges are more closely grouped together and nodes with no edges between them are further away from each other. Using color codes (in such a graph) one can signal a conflict between two norms. To make this graph interactive, the user can click on a norm (a node) or an edge to get further details and insights. Furthermore, the layout of the graph can be changed in such a way that nodes (norms) that have the same property (for example agents that have to adhere to it) are closer together and others are further away. This dynamic norm graph would (visually) provide deeper and more detailed insights into the connections between the norms.

5.3.3 Interpretation & Discussion

To interpret the results and to relate them back to the original question proposed in Section 5.2.2.1, *what the policymaker wants to see*, we have to make a differentiation between norm conflicts that can happen, based on fulfillment/violation condition conflicts or potentially conflicting situations but **have not happened yet** in the simulation run, and norm conflicts that *have happened* in the simulation. We call the former **ex ante** (before the norm conflicting situation occurred in the simulation run), and the latter *ex post* (after the norm conflicting situation occurred in the simulation run). Figure 5.8 shows this divide. Furthermore, the figure shows a differentiation between norms that conflict based on their fulfillment/violation conditions and norms that do not conflict in themselves but can create conflicting situations.

The results of the focus group study reported in the previous section fall into the *ex post* (after the conflicting situation occurred in the simulation run) category, such as an agent has to decide between speeding or being late for a meeting. When presented with such a situation, the participants had a very clear idea on what they wanted to see and how to analyze it. In our example, the drinking and consequently being late for a meeting the next day conflict, it was clear for people to look, at for

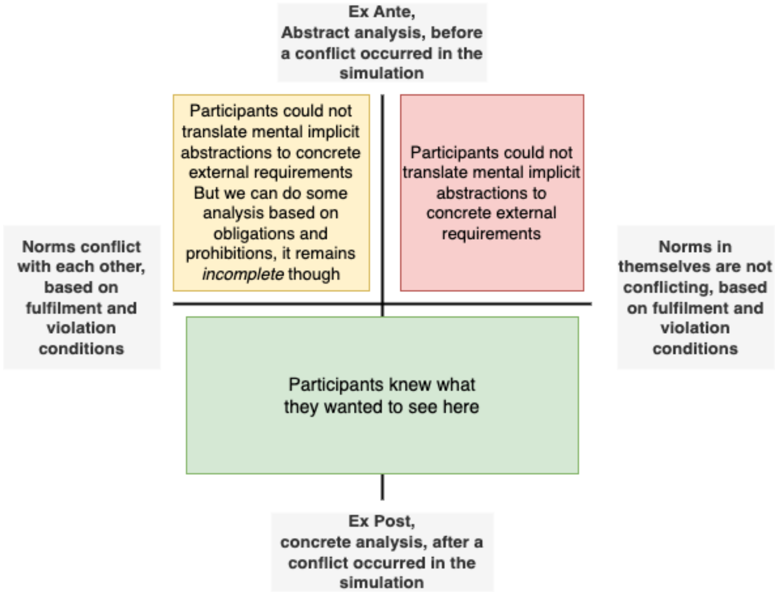


Figure 5.8: Two-dimensional analysis space with color coding, from has been addressed (green) to has not been addressed (red) in the focus group study, taken from our paper [79]

example different neighborhoods or use filters for agents. All these things were meant to provide them with deeper insight into the conflict and how to find measures against it. Even the dynamic norm graph, only helps in terms of an overview, but does not provide deeper insights into the concrete reasons why a norm would potentially be violated. Furthermore, it can be very problematic to highlight potential norm conflicts in the dynamic norm graph if the norms are in themselves not conflicting (red area in Figure 5.8). This is then very challenging for the coloring of the edges in the dynamic norm graph.

Based on the overall results, we present potential solutions for the *ex ante* category in the next section and take inspiration from the discussed onClick agent inspection, as this function can be very useful. As for the other elements mentioned in the results, existing work, such as our ASSOCC project [37], can be used as a basis for further development.

5.4 A potential Interaction Tool

The results in the previous sections show that there are many approaches and tools, see for example the visualizations in the ASSOCC project [123], for the data analysis and broad analysis of a simulation. This means that for the green rectangle presented in Figure 5.8, we (the modelers) have the tools and the knowledge about what is required there. However, there are no tools to support the yellow and red areas shown in Figure 5.8, especially when it comes to norms.

These are the problems that we tackle in this section and present our preliminary work on them. To do this, we firstly split the interaction with norms into three phases, 1) the norm modeling phase (mainly done by the users), 2)the norm implementation phase (mainly done by the modeler), and 3) the norm monitoring phase after the implementation (mainly done by the user). For each of these phases, we will present one potential tool to support this phase. An overview of this three phase process and the connected tools can be seen in Figure 5.9.

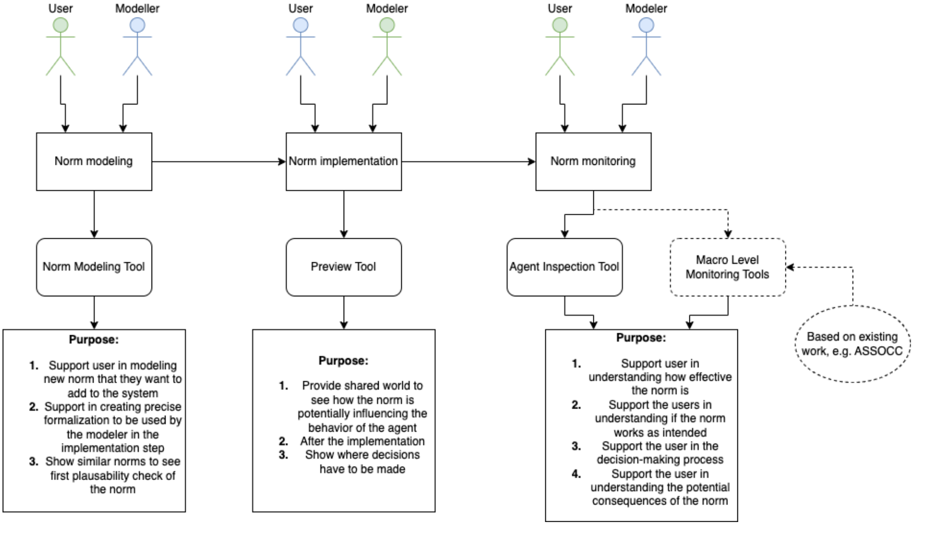


Figure 5.9: Three different phases of interacting with norms. The green color highlights the lead role in this step and the blue color the supportive role. Line represents other work for inspiration but is not further focused on in this thesis.

5.4.1 Norm Modeling Phase

The first step in the process is the norm modeling phase. The norm modeling phase targets two processes done by the user: 1) modifying an existing norm, and 2) adding a new norm. Supporting the user in these two processes is the purpose of, what we call, the *norm modeling tool*. The main challenge is to enable the user to add a new norm or modify and existing one in such a precise way that the modeler can use the desired specification to make changes on the code level of the simulation, e.g writing the code for a new norm. To achieve this complex task, the tool guides the user through the process to breakdown the different steps, similar to a guiding dialog, and inspired by the the scenario selection cards from the ASSOCC project [123] (Figure 5.6a).

The first step is to select between adding a new norm or modifying an existing one Figure 5.10.

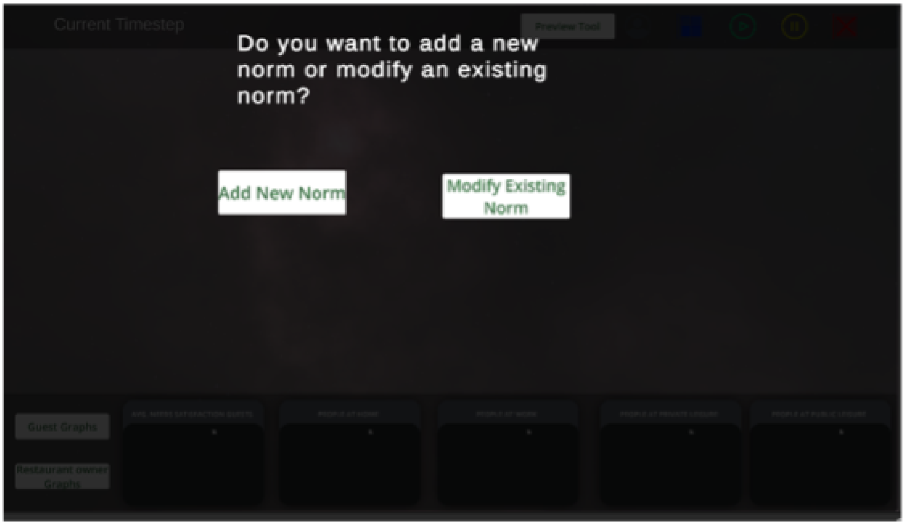


Figure 5.10: Beginning selection screen norm modeling tool

When the user decides to **modify an existing norm**, the following screen in Figure 5.11 is displayed. In this first step, the user can select a norm that they want to modify. To select a norm, the user has multiple options. They can either search a norm by name, or purpose (i.e. the intention of the norm[1]), or use other filters, such as if the norm is currently active or which agents have to adhere to the norm.

After the user decided on a norm, they are presented with another screen, see Figure 5.12. This is the screen where the user can actually modify a norm. Here the norm is displayed with all the different elements. The ones that can be modified have a button for the user to click on, indicating their modifiability. Other elements, such as the affected needs or the deontic part of the norm cannot be modified, as they either depend on other elements from the norm, or are simply set in the code to ensure the code's functioning.

To support the user and prevent inputs that would potentially break the simulation, some basic validation can be used to validate some input fields. For example if the punishment is a monetary fine, the input field can ensure that only numbers are entered and not words. When the desired changes are made the user can safe the modified norm and continue with the simulation.

This process has many challenges. The biggest challenge is to decide what parts of norm can be modified by the user. Some elements might be more easily modifiable than others. For example, changing the amount of the fine (punishment) is a minor change in the sense that it can be easily done from an implementation viewpoint as only the value of a variable changes, e.g. from 30($) to 40($). Changing the type of punishment however, from a fine to something else (e.g. a prison sentence or the

[1]Explained in the following paragraphs

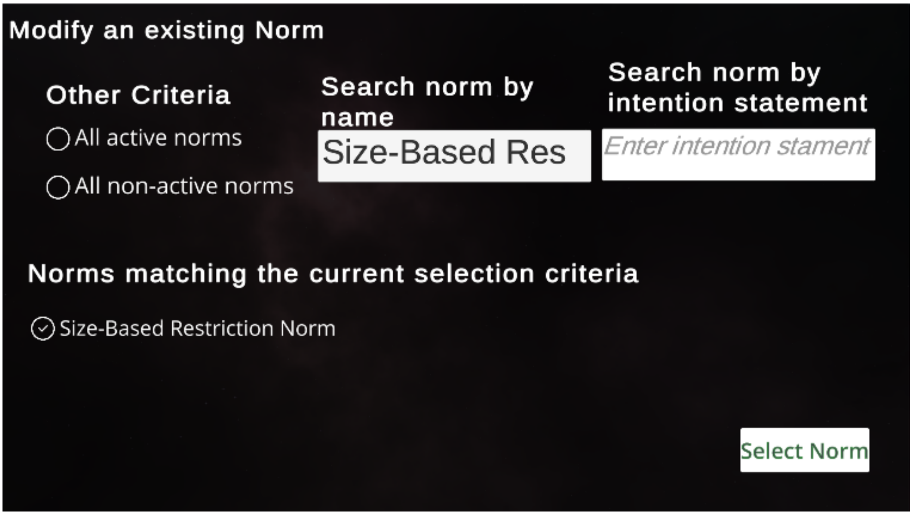


Figure 5.11: Screen for selecting a norm to change. The elements are button to show that it can be clicked on. Some buttons are grayed out to show that they cannot be modified by the user.

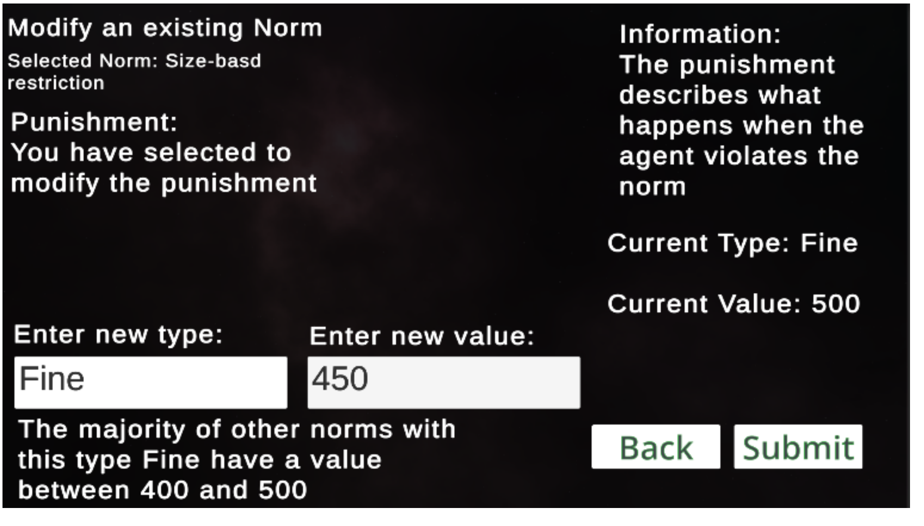


Figure 5.12: Screen for modifying the selected norm.

removal of a driver's license) might not be as trivial, because the underlying type might change,, e.g. a numerical type in case of the fine might change to an general text field for the name of the action to go to prison. This is not a trivial change from an implementation point of view. Even more so, some elements of the norm, such as the deontic part (e.g. must or must not), should not be allowed to be changeable as they are crucial to the purpose of the norm and the functioning of the code. When looking at the size-based restriction for example, changing the deontic element from must not to must results in the complete opposite meaning of the norm, and would thus contradict the purpose of the norm.

Regarding the purpose of the norm, the question also is if that should be changeable. To some extend this might be desirable, if new avenues for the use of the norm can be found. Going back to the size-based restriction norm as an example. The initial purpose (intention statement) might have been with regards to the COVID-19 pandemic. Now the norm might have been proven useful and the user wants to use the norm in other situations as well. Of course this comes down to the discussion about the use of simulations in general, as a specific simulation is usually only developed for a specific purpose, meaning that the intention of the norm might only be alterable to a certain extend, as the scope of the simulation is limited. To solve this, it might be relevant to investigate the potential use of norm configuration files, meaning that the user can export the norms they used in a simulation to an external file, and then import it again for a different simulation.

When the user decides to **add a new norm** the next step is not to directly jump into the formalization of the norm. This would be too overwhelming for the user, given that they are not well-versed in the precise formalizations needed for a norm to be used in the simulation model. To avoid this, we propose to start with the intention that the user has for adding a new norm. They have a reason for adding a new norm. For example, when adding the size-based restriction norm the people have in mind to add a norm to limit the number of people in the restaurant. This is what they formulate in the intention statement. The intention statement is used to find similar norms that can be used as examples for further norm specification. While in Figure 5.13 the intention statement input is still one field, it can separate to different fields to make the word extraction easier.

Once a similar norm is found and the intention is clear, the user can input the different elements of the norm. Figure 5.14 shows this process. The left side shows the new norm and the current specification of it, and the right side shows the similar example norm for comparison and guidance. The middle shows the current element for input and a short description of it.

Finally, after the new norm has completely been specified the user is prompted with the option to have a preview of the norm in the preview tool (see the next section, Section 5.4.2) before adding to the simulation after the norm has been coded in the simulation by the modeler, see Figure 5.15.

Within this process, there are some key challenges however. The first one is to identify what exactly similar means. Similar can be either words, such as restaurants, or the purpose of the norm (i.e. the intention statement). Here generative AI approaches, such as Large Language Models (LLMs) can be used to identify those similarities, as

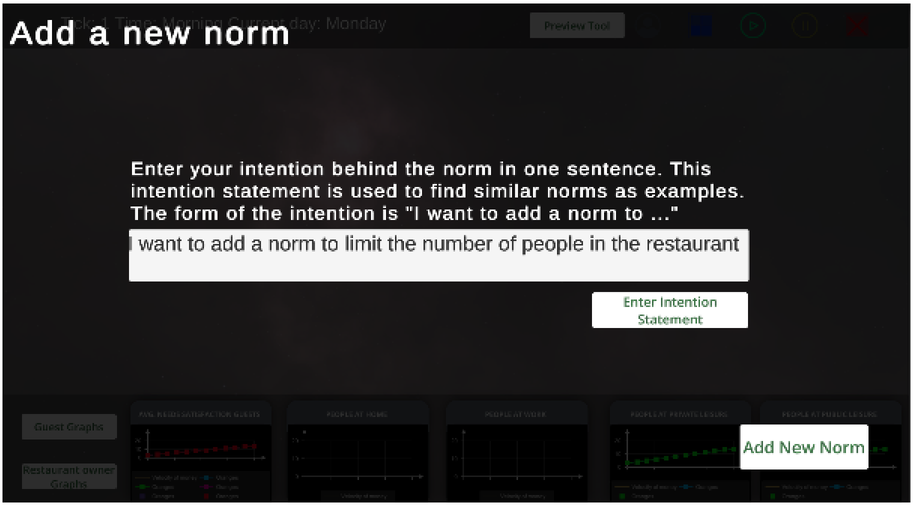


Figure 5.13: Intention statement input field

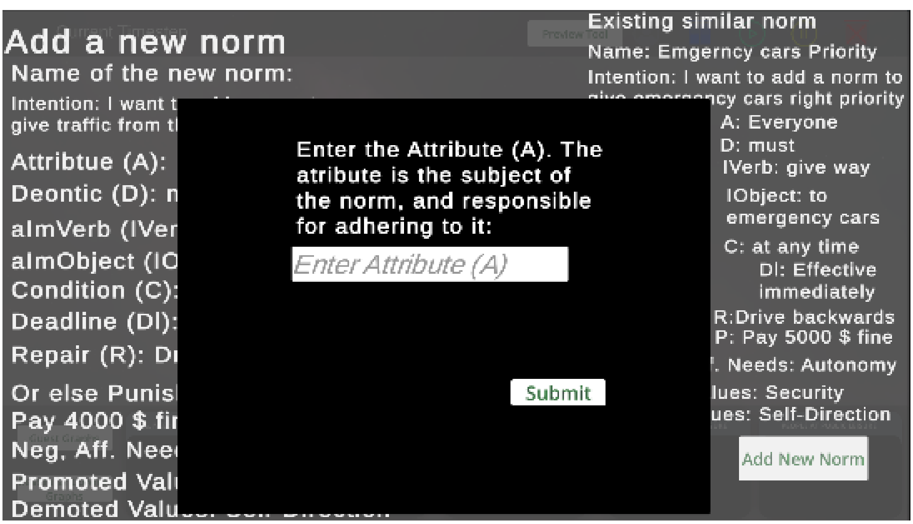


Figure 5.14: Norm input field

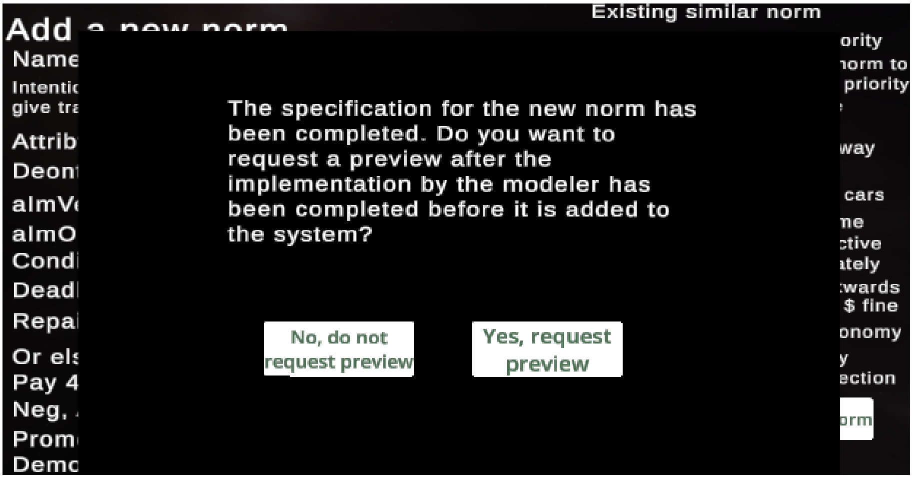


Figure 5.15: Final screen norm modeling, adding a new norm

they are suitable for such tasks [51]. It is important to provide guidance and examples for the user for specifying new norms, as it can reduce the risk of inputs being outliers, where being an outlier can mean something like a fine being way too low and way too high with respect to other norms. Of course this depends strongly on the simulation and is only to be seen as a rough guideline.

The second challenge is to identify what can be specified by the user. Some norm elements, such as the negatively affected needs, and pro-/demoted values, are strongly dependent on the other elements of the norm, and very hard for the user to identify. This means that those elements should come from the social simulation system, after the other elements have been specified by the user. For example, the norm that the use of social media is forbidden at the workplace during working hours is negatively impacting a relaxation related need of the workers. If the norm is reversed, i.e. people must use social media for one hour during working hours, a relaxation related need is now not negatively impacted anymore, but potentially other needs related to work productivity. This example shows that it is very challenging, and not a trivial task for the user to do, but rather that the system needs to support that process.

Finally, this is all very text-based and overwhelming right now, and allowing for all inputs. There needs to be ways to restrict the input in a way (not too restrictive) to only for values that make sense for the specific element. This also then has the inherent challenge of what makes sense for a given input field and what not. To improve the usability searchable dropdowns could be used instead of pure text-based input. This could then also help with limiting the input possibilities of the user. Here, further focus group studies can be beneficial to further identify what exact support is needed, and what options are preferred by the potential users of the simulation.

When specifying the elements it would of course be beneficial to signal to the user that the current specification is creating normative conflicts, such as having to be at two different places at the same time, or inconsistencies, such as an action simultaneously being obliged and prohibited [152]. However, this is not a trivial task, as this can be very situation specific for some norms [63]. For example, the norms of giving traffic from the right the right of way, and giving emergency cars the right of way, are in themselves not conflicting. However, when suddenly an emergency is coming from the left (having the point of view of an ordinary car), a conflicting situation occurred, and decisions have to be made how to resolve that conflict. Since this is a very complex dynamic, this needs to be supported when implementing the norm.

5.4.2 Norm Implementation Phase

While the implementation of a new norm seems straightforward, i.e. the modeler uses the specification from the user and writes the code for the model accordingly, there are still many things that need to be taken care of, and that can be supported. These can be for example norm consistency checks, as mentioned in the previous section, or to see if there are potential conflicts with other norms that are not obvious or can produce conflicting situations within themselves.

To make this more clear, we want to use the following two norms (formulated more colloquially) in this section as examples: N1) Traffic from the right has the right of way, and N2) Emergency cars have the right of way. While in itself these norms seem fine, there are many potential conflicts that can arise, such as the ones that we will discuss now.

The first potential conflict comes when considering the two norms together. Let's say we have a situation where one (normal, non-emergency) car is coming from the right, and an emergency coming form the left at a crossing. The obvious choice for us, as humans, is to s let the emergency cars pass first, as they take precedence in this case, i.e. in our mind, N2 takes precedence and is overruling N1. However, the agents in the simulation do not have this intuitive thinking to make this decision. The modeler needs to provide the agents with the information for this specific situation (and similar ones) what to do, e.g. N2 overrules N1. But this is not a decision that the modeler should make by themselves. The user should be involved in this decision and (preferably) decide what to do.

The second potential conflict arises just from N2 in itself. It can be possible that just with this norm alone conflicting situation can arise. For example, it can happen that two emergency cars are at the crossing instead of one non-emergency car, and one emergency car. The question here now is which emergency car takes precedence. One could say an ambulance is more important than a police car for example, but this might be highly subjective. Again, a decision must be made and integrated into the system by the modeler who is guided by the user in the decision.

To support the modeler and non-expert user in this process, and tackle these challenges, we propose the, what we call, *preview tool*, as mentioned in Figure 5.9. The purpose of the preview tool is to provide a shared understanding of the simulation for the non-expert user and the modeler to identify potentially conflicting situations, and to see if the norms work the intended way in isolation. We envision the preview tool to be essentially a mini simulation within the overall simulation to look at specific situations. With this, the modeler together with the user can simulate and investigate potentially conflicting situations and decide how to resolve them.

An example situation based on the two norms N1 and N2 can be seen in Figure 5.16. It shows the starting situation in the preview tool. A crossing with a non-emergency car (cyan car), and an emergency car (represented by the police car in this case). This situations seems to be floating in the middle of nowhere. This is intended to show that this is happening in isolation to focus on the specific situation. It is important to note here that what is happening in this preview tool has no direct influence on the overall simulation (except the follow-up decisions that are made based on it). The top right of the preview tool contains the norms that are currently included in the preview tool. They can be toggled on and off, as indicted by the check mark. Furthermore, a dropdown is present so that norms can be dynamically added or removed from the current preview. Finally, the two button in the top left are a start button to start the preview run, and a back button to return to the overall simulation.

Figure 5.16: Starting situation

After pressing the start button and since the the police car is coming from the right, we can see in Figure 5.17 that there is no conflict, as the emergency car (police car) is coming from the right, and the non-emergency car (cyan car) waits according to the norms N1 and N2. But this not the only situation that can occur. So for the modeler and user to explore potential situations, the preview tool that we envision has a drag and drop functionality. This allows the cars to be dragged to different locations on the road to explore the potential behavior. This can then result in the situation shown in Figure 5.18 where the non-emergency car (cyan car) is coming from the right of the emergency car (police car). This creates the conflict between the norms N1 and N2, as discussed in the previous paragraphs, which is now perfectly visible and obvious for the modeler, and user to see. This enables them now to discuss the situation and make decisions to resolve the conflict. This shows that the preview tool also create a shared view and understanding of what is going on in the model. This can then help to bridge the gap between the two worlds shown in Figure 5.1.

Figure 5.17: Situation with no conflict. Emergency car coming from the right

Figure 5.18: Situation with conflict. Emergency car coming from the left

5.4.3 Norm Monitoring Phase

The final phase for norms (see Figure 5.9) is the norm monitoring phase. The norm monitoring phase comes after the norm is implemented. The purpose of this phase is to analyze and observe how the norm is being used by the agents and the potential effects it has. This of course also includes showing that the norm is being used at all. It is important to show that because the implementation stage only focuses on the actual implementation process of the norm and writing the code for it with the necessary decisions relevant for that process. It is not guaranteed that the norm is used by the agents. However, this is crucial for building trust with the non-expert user (policymaker) of the simulation, as they need to see that the norm is really used by the agents in their decision-making process.

To do this, we envision a, what we call, *agent inspection tool* which is focusing on a single agent[1]. With this inspection tool the user of the simulation will be able to see the current norms that the agent is taking into account and that are affecting their decision-making. This has not been done before. We acknowledge that NetLogo [162] has an agent inspector functionality. However, this was developed for debugging purposes, and has therefore some major drawbacks.

Figure 5.19 shows a screenshot taken from the NetLogo interface of our ASSOCC project [37, 123]. We can see in the figure that it is possible to inspect an agent, here we took a screenshot of a random worker agent. Furthermore, the green circle in the top center on the map shows that there is some kind of focus that can be adjusted by the slider in the inspector, as displayed on the right side of the figure. However, there are major drawbacks. The first one is that the inspector only shows the variables with their names and values, as further focused on in Figure 5.20. This is it not very intuitive to understand what is going on, as it strongly depends on how the variables are named. Furthermore, it shows all the variables, and the user has to scroll through the list to find the one that they are interested in. Even then this might not be very helpful as only a number or a string of words is displayed without further information about the meaning of that.

This also leads to the second problem that the agent inspection tool is not adjustable in its size. While the triangles in the in the top left corner of the small map and the variable list allows the user to minimize that part, it also shrinks the size of the overall inspection view, meaning that the actual size of the variable list (for example) does not change, and that the user still has to scroll to find the desired variable.

To overcome these challenges, and provide more insights to the user, we propose a novel agent inspection tool shown in Figure 5.21. The left side (light gray show the list of existing agents in the the simulation as buttons. When clicking on an agent button, the button is highlighted as selected, and the detailed agent information (in front of the dark background appears). This list of all agents could filtered further or the non-expert user could click on an agent on the map (if agents are not teleport during timesteps, as in our ASSOCC project [37, 123]).

The left side of the dark background area (in Figure 5.21) shows information about

[1]For broader macro level tools, we can use existing work, from e.g. our ASSOCC project [36], as shown with the dotted lines in Figure 5.9

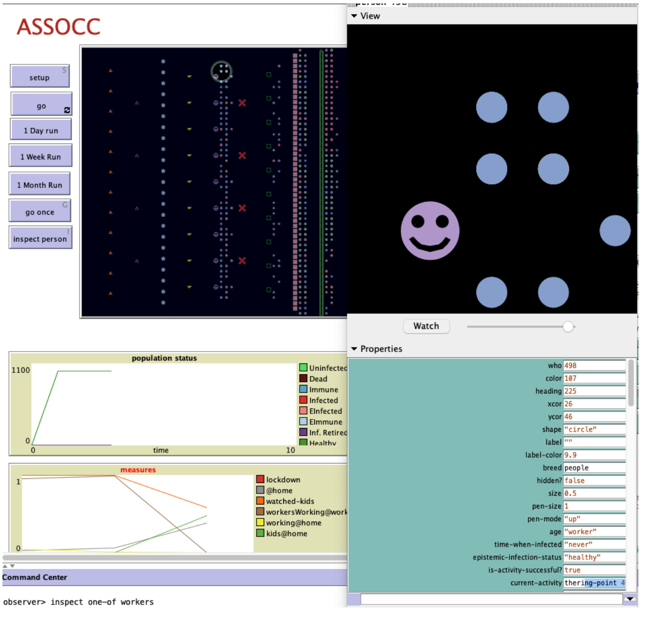


Figure 5.19: NetLogo agent inspector (on the right), taken from our ASSOCC project NetLogo interface [37]

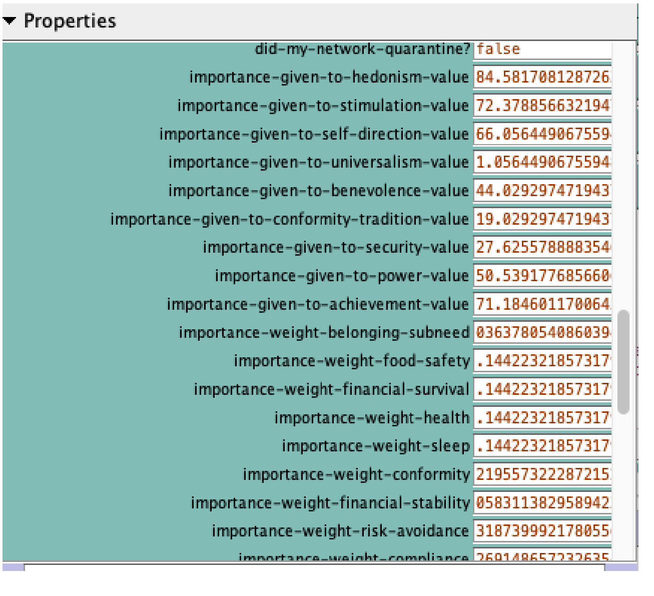


Figure 5.20: Variable focus of the agent inspector, taken from our ASSOCC project [37] NetLogo interface

Figure 5.21: Proposed agent inspection tool

the agent, such as their perspective, and characteristics. Below, the needs of the agent can be found with their current satisfaction and the highlighted threshold (vertical line). Red means that the need is not satisfied, and green means that the need is satisfied. The button below the needs can be used to gain more information about the needs, for example by textual descriptions about them or in other forms. These description can include the need means or represents, and which action can be used to satisfy the need or are negatively affecting the need.

The top middle part shows the current action that the agent wants to take. Again a button is there to provide more information about the action, such as which needs are positively and negatively influenced by this action and which norms this action is adhering to or violating, to provide a better understanding for the user why this action has been chosen. Here the research on explainable BDI agents by Broekens et al. [14] and Harbers et al.[64] could be used by providing explanations in the form of: because of $< needs >$ and $< norms >$, we chose $< action >$.

The most important part in terms of norms is the right side which lists the current active norms that the agent is aware of. Each norm in the list is represented as a button to indicate that clicking on it provides more information. Another way to get more information can be when clicking on a norm or a need to then show the negatively impacted needs (when clicking on a norm) or the norms which negatively impact the need (when clicking on the need). This can also provide a better understanding for the non-expert user.

A challenge that comes with this inspection tool however is the complexity and amount of information imposed on the user. On the one hand, it might be good to have this dashboard-like quick overview with all information to make a quick assessment of what is going on insight the agent. With this, the user could quickly look and see what the current state of the agent is, what action they are going to take and which norms are used by the agent. Furthermore, the buttons can be used to hide more complexity behind them and having more information in the back.

On the other hand, presenting information like this to the user can also be overwhelming. When looking at the dashboard-like information it might not immediately be clear where to look at for the non-expert user or to understand which part is the most relevant at this point in time. Furthermore, it might not be the optimal way to show which norms are considered by the agent. While showing that the agent is aware of them (by being in the list of currently considered norms) is a good start and one would assume that being aware means to take it into account, there should be more research done how this can be improved.

Another challenge is to make the user aware that such a tool exists in the simulation. Unlike the other two tools from the previous sections, the agent inspection tool is not connected to a specific action of adding or modify a norm. The user can inspect an agent at any point during the simulation run. So the question is if it is enough to have a button or icon in the overall simulation interface that represents the possibility to select an agent or if further hints, such as notifications, should happen during the simulation run to make the non-expert user aware of this possibility of inspection. However, one has to be careful here to not overwhelm and distract the user with notifications. Furthermore, this then raises the question about the triggers of these notifications, e.g. random times during the simulation run or in regular intervals at ever specific timestep, or other triggers based on the agent's behavior.

5.5 Conclusion - A new research agenda

The results and research that we presented in this chapter show that we (the social simulation community) are far away from an actual interaction tool that provides the necessary capabilities for a fruitful interaction between non-expert users and modelers with the every increasing (by necessity, as shown in this thesis) complexity of models and agent deliberation architectures.

While approaches and advancements, especially our ASSOCC project [37, 123] have been made, they are not sufficient, as they cover only parts of the necessary functionality in terms of general analysis of the simulation. What we (the modelers as well as the users) need are interaction possibilities that have a special focus on norms, and not just a focus on the overall simulation, thereby leaving the user to guess where and how norms affect the decision-making process of the agent.

One could now suggest to use participatory modeling approaches to tackle these issues. Such approaches have been used in the social simulation community very early on (see e.g. Ramanath & Gilbert [126] or very recently (see e.g. Ahrweiler [1]). However, participatory modeling approaches focus on a different part of the modeling process. Melchior et al. [100] show that participatory modeling is very fruitful in building the simulation model, as shown in their recent collaborative ABM policy framework [100]. Meaning that participatory modeling can be used to identify the perspectives on norms relevant for the model, what their needs and potential (typical) actions are, and so forth. But participatory modeling is not suitable for later support in the simulation. These approaches *do not* focus on interactions to make the results of the simulation and the behavior of the agents more accessible. For example Ahrweiler et al. [1] also pointed out in their work: "[...] new visualization and interactive technologies can help to make simulation results more accessible to stakeholders" [1, p.8].

This is exactly where our preliminary results come into play. By making the norms modifiable for the user, the user is in themselves more strongly *integrated* in the use of the model, and not just a simple observer. To properly support this increased level of integration, the user needs to be properly supported. This support is not only focusing on the modifiability of the simulation, but also includes the communication wit the modeler to enable an efficient and meaningful use of the simulation, and collaboration with the modeler.

For this support later in the simulation, we (the social simulation community) need novel interaction and communication tools, especially with a focus on norms, as identified by the challenges in this chapter, and the results of the focus group study. Furthermore, this interaction (with the simulation and the modeler) needs to feel natural and as close to the human mind as possible to be fruitful [79] for the non-expert user.

To contribute to the development of and the research on these novel interactions tools, we divided the process of norms into three stages: the norm modeling stage, the norm implementation stage, and the norm monitoring stage (see Figure 5.9). For each of these stages we proposed a potential solution and first step on how a tool for this process could look like: the norm builder tool (Section 5.4.1), the preview tool (Section 5.4.2) and the agent inspection tool (Section 5.4.3). However, these are just first steps on a long path.

To proceed from here, further user studies have to be performed to see if the tools that we proposed are actually useful for the users and reflect what they want and need. Additionally, the tools we propose need to the implemented in several models to identify their potential usage and pitfalls that need to be solved. The development of such tools is only at the beginning!

Furthermore, we find a recently emerging strand of research promoted by Schimpf & Castellani [134] promising, called "approachable modeling and smart methods – AM-Smart for short" [134, p.1]. The authors argue to consider more cognitive sensitive approaches and dimensions. Such AM-Smart platforms can then "[...] help users to simultaneously use and learn new methods"[134, p.2]. Another key point by the authors is to leverage visual reasoning skills. Also, their promoted strand of research aims to enable a seamless, and responsive way to gather and analyze information. Finally, such platforms "specifically target learning through doing, as they facilitate *user-driven* learning of a topic, primarily through intuitive, tailored supports[...]"[134, p.2]. These are all factors and elements that are relevant for an interaction tool for supporting non-expert users. Consequently, we argue that this is a highly relevant research area to consider in future research.

Finally, while advances in generative AI, such as LLMs, have to be considered and monitored, we argue for caution given their potential pitfalls, such as hallucinations (making up content that is not existing) or gaps in logical reasoning. This means that they might be suitable for finding similar norms that can be used as references when adding or modifying norms, as discussed in Section 5.4.1 or when using simpler agents with not many parameters to consider. However, for complex agents LLMs are not suitable, as they will inevitably fail to provide a concise reasoning for the agents.

Chapter 6

Conclusion & Future Work

In this thesis, we set out to create social simulations which show more realistic normative behavior to support policymakers in their decision-making process. In the real world people react with a diversity of responses to norms (obeying, violating, and circumventing the norm) which is rarely reflected in simulations that aim to support policymakers. Increasing this part of social realism is crucial, because only then social simulations can utilize their full potential as a powerful tool to support policymakers in their decision-making.

Our goal is to create social simulations that provide an in silico test bed for policymakers to investigate the potential effects that their desired policies might have. In order to achieve this goal we need to increase both norm realism in agent behavior as well as increase the usability of the simulation. Norm realism and usability are tightly connected, as policymakers need to understand *why* things are happening in the simulation, and *why* the agents behave the way they do. Only then can policymakers engage with making potential changes, and see if their intended policies have their intended effects in the simulation.

These two crucial elements (norm realistic behavior and usability) are reflected in the challenges described in the introduction (of this thesis), and captured in the following ***main research question (Main RQ): How to increase norm realism and usability of social simulations to support policymakers in their decision making?*** In the following sections we discuss how we have answered this research question as well as the more concrete sub questions that we posed in the introduction.

6.1 Norm Realistic Behavior

The **first key element** for dealing with norm realism is that people focus only on the parts of the norm that are relevant for them [82]. This means that they interpret a norm differently, based on what they perceive it means for them. E.g. a speed limit for a driver might mean to check whether there are cameras or police before deciding on For a police officer, For a police officer, it might mean avoiding roads with strict speed limits, as they could require frequent enforcement and issuing many fines which would result in extra paperwork and time spent issuing these fines rather than focusing on other offenses. We call this subjective reasoning a *perspective* on norms. Furthermore, it is crucial that a perspective on norms allows for the various norm reactions that exist: obeying to a norm, obeying and circumventing a norm, and violating a norm [78, 81, 82].

Formalizing different interpretations of norms (*RQ1*[1]), requires a way to capture and transform the objective world, which is the same for everyone, into a perceived subjective world view. To achieve this, we used the social constructs of needs, values, and social affordances in our definition of a perspective on norms (see Definition 1 in Section 3.3 in Chapter 3).

The needs are crucial as they reflect the current state of an agent (person in the simulation). This is the key for the decision-making of the agent, as they determine the current attractiveness of an action for an agent to for example obey a norm or violate it. For example, if a restaurant owner feels they have enough money and thus their financial needs are satisfied, that person is more likely to simply obey the new norm of limiting seats in the restaurant compared to a person whose financial needs are not met. This person will then in turn do more likely something to avoid the new norm negatively impacting their financial needs.

While needs provide an individual (micro level) guide of a person's behavior, the behavior of a person is also guided by the society or group (macro level) they are a part of. This is where values come into play. The values of a society influence what is important for an individual (i.e. influence the importance of their needs), and thus also the attractiveness of normative behavior. For example, if a society values norm compliance very highly, then the need of complying to norms (compliance need) of a person is also in general quite high, and thus norm complying behavior is more likely for that person.

In order for a person to determine which courses of action are available to them, they need to understand which actions are possible with a subjective view on the physical objects in a shared physical world. This is where the social affordances come into play. They describe the social action possibilities provided by a physical object, and thus what social actions are afforded by the physical object. This is very important, as based on their perspective, a person has different actions available with different meanings available for them. For example the physical object of a taxi affords different actions to the taxi driver, such as driving or work, compared to the passengers of the taxi, e.g. transportation.

In our simulations norms must also be understood relevant to these different perspectives. This requires norms to be more complex, detailing their impacted needs and targeted object. To enable subjective reasoning about norms, we proposed the ADICDlRO (see Section 3.4) framework of norms, which highlights the norm's object and personal significance. This allows agents to assess potential impacts on their lives. Identifying negatively impacted needs is crucial to determine if a norm poses an immediate threat. If these needs are satisfied, the agent perceives no urgent risk

[1]RQ1: How can different interpretations of norms be formalized?

and proceeds normally, following the norm. When the needs of an agent are no longer satisfied or are threatened to be not satisfied by following the norm, the agent will deliberate about violating the norm, or take actions to circumvent the norm and counteract the perceived negative impact of the norm on their needs. Values play an important role in this decision as well. Promoted and demoted values further help resolve norm conflicts, guiding agents in choosing which norms to follow or disregard if not all of them can be satisfied at the same time.

The **second key element** for norm realistic behavior in social simulations is how to integrate the formalizations of different perspectives of norms in the agents of the simulations that are capable of dealing with different perspectives (*RQ2*[1]). To do this, we developed a novel agent deliberation architecture capable of dealing with different interpretations of norms, called the Perspective-Based Agent Deliberation Architecture (PBADA) in Chapter 3 in Section 3.5 which is based on a two-tiered deliberation process: a base level and an action deliberation level.

The base level stores and handles the perspective specific elements, such as the needs of an agent and the available actions. Furthermore, the base level handles subjective updates to the agent's worldview and personal state, including needs and expected need satisfaction gains of their actions. This allows for the integration of different perspectives on norms into the deliberation process of the agent, as they are now building the basis of the decision-making process in the action deliberation level. Additionally, this means that only the elements in the base level (which represents the different elements of a perspective on norms) need to be instantiated differently for each perspective in the simulation, while the action deliberation process is the same for all agents.

The action deliberation level determines the agent's course of action based on their current context and inner state. Not only the state of the agent is reflected but also the normative context of an agent. This is crucial, as it allows the agent to reason about how norms potentially impact their further decision-making. To do so, norms are treated as explicit objects within the architecture enables agents to reason objectively about norms and decide whether complex deliberation is required. This differentiation was essential for achieving clear and explicit reasoning about norms. It also shows that norms come into play at various stages in the decision-making process, and not only in the action selection process. They also play a role when deciding how complex the deliberation has to be, i.e. what deliberation level to chose: the habitual action without further deliberation (habitual deliberation), reacting to the behavior (caused by norms) of other agents (medium complex deliberation), complex deliberation potentially caused by a norm negatively impacting salient needs of the agent.

The use case of our PBADA agent deliberation architecture (see Chapter 4) provided support that our architecture is indeed showing norm realistic behavior in line with what we see in reality. Therefore we could at least partially verify and validate our PBADA architecture. This provides support for our approach on different perspectives on norms and the resulting agent deliberation architecture.

[1] RQ2: How does an agent deliberation architecture need to look like to integrate different perspectives on norms?

6.2 Usability of the simulation

The second part highlighted in the main research question, and outlined in the challenges in the introduction, is the usability of the simulation. Usability of the simulation means that the policymakers can modify the norms in the simulation and add new ones (*RQ3*[1]), as well as empowering the policymaker (non-expert user) to understand the ensuing changes in the result of the simulation (*RQ4*[2]). Identifying the support needed for policymakers to modify existing norms or add new ones on the fly goes hand in hand with their ability to use the simulation in a for them meaningful way. They can only modify a norm, if they can understand the potential consequences of that. By potential consequences we mean that policymakers need to be able to understand how the norm they added or modified is used by the agents, how the norm is impacting their needs and actions and how their behavior changes as a consequence of that. Policymakers can only make decisions on testing various norms (policies) and use the results of the simulation, when they know and understand what is going on in the simulation, and why it is going on the way it does. They both are the pre-requisites and consequences of each other.

In general, to understand the support necessary for the policymakers, it is important to see an interaction tool as a communication tool between the policymaker and the modeler. The policymaker is responsible for specifying the norms they want to have in the simulation and to observer their potential effects, while the modeler is responsible for implementing those norms, and resolving (implementation) conflicts between or within the norms. This means that the modeler will not be obsolete but rather the communication between both the modeler and the policymaker is supported and fostered.

This can be achieved in a three step process (see Figure 5.9 in Section 5.4 in Chapter 5). The first step is the norm modeling step, where the desired norm is modeled (norm modeling tool). The second step is the norm implementation step, where the modeled norm is implemented in the simulation and potential conflicts are resolved (preview tool). In the third and final norm monitoring step, the user (policymaker) can then observe and analyze the simulation on an agent level (agent inspection tool) and simulation level.

The norm modeling tool (see Section 5.4.1 in Chapter 5), is responsible for supporting the policymaker in adding new norms to the system or modifying existing ones. The tool provides this support by splitting up the process into several distinct steps: selection of adding a new norm or modifying an existing one, searching for existing norms by name and intention statement (in the case of modifying an existing norm) or adding a new norm by first formulating the intention statement, and then for each case (modifying an existing norm or adding a new norm) showing the detailed input fields for entering the specific values of the elements of the norm with examples of similar existing norms to provide guidance. Doing it like this does not only provide the user with support by showing how similar norms are specified in the simulation, but also

[1]RQ3: How can users be supported to change norms and add new ones during the simulation run?

[2]RQ4: What support do non-expert users need from a user interaction tool to be empowered to use the simulation in a for them meaningful way?

reduces the complexity of the process by splitting up the decision steps.

The preview tool (see Section 5.4.2 in Chapter 5) provides a mini simulation inside the overall simulation. This supports the user as well as the modeler to see if the norm in isolation works as expected, as well as to see if there are potential conflicts between two or more norms in certain situations, or even the same norm. The isolated environment prevents any conflicts with the existing simulation, as well as allowing for a focused view on the norm(s) in question. Furthermore, this tool supports the communication between the policymaker and the modeler by creating a shared mental model of the simulation and the elements the simulation entails, and a shared basis for discussion.

The final tool, the agent inspection tool (see Section 5.4.3 in Chapter 5), enables policymakers to inspect single agents and understand how they make their decisions at certain points in time. This is crucial for understanding how a norm is processed by the agents, and why the agents, for example, decide to violate the norm. These insights can then be used to identify further changes that might be necessary to control this norm breaking behavior of the agents.

6.3 Future Work

Given the nature of our research, and as we have pointed out throughout this thesis, there are many courses for future work to be done. They can roughly be divided into the following two groups: work on the formalisms and the agent deliberation architecture, and work on the usability and non-expert user support. We only provided the first initial steps and many more have to follow.

The **formalisms of different perspectives on norms and the subsequent agent deliberation architecture** can be extended in many ways, as we have discussed in Chapter 3. The first way is to include more motivators and social theories, such as culture. Culture influences our values, acts as a macro guide for our behavior [37, 52, 88, 150], and can play a (crucial) role in our behavior when responding to norms, as shown for example in our ASSOCC project [52, 88]. With the integration of culture and the connection to the values of the agent and their needs (e.g. the influence on the importance of certain needs, such as compliance, as we did in our ASSOCC project [37]), it would be possible to situate different perspectives on norms in different cultures. Another aspect of future research related to the values is the change in priorities of values over time. In its current form, we assume that the priority of values ($\mathcal{PRIOV}$) of the agent is static over their lifetime in the simulation. However, in real life this is not the case. We (people) value different things higher when we are younger compared to when we are older. Integrating this into different perspectives on norms depends however strongly on the purpose of the simulation. While in general it might be interesting, it is not really useful if the simulation length does not incorporate a long enough time range. For example, in our ASSOCC project [37], we only simulated a couple of months and thus it was too short of a time span to account for a change in priorities of values.

Another aspect in this area of research is the formalism of a perspective on norms itself. So far we consider them static, and one agent has one perspective over the course of the simulation, for example being a restaurant owner. But this might not be the case in reality, as we (people) have many perspectives at the same time, but only one of them is currently active. For example, the restaurant owner could decide one evening to go out to eat in another restaurant to either look what potential rivals are doing or just to enjoy a different meal outside. This would then make the restaurant owner a guest with a different perspective on norms, and also subsequent actions available. They are still the restaurant owner of their own restaurant but also a guest in a different restaurant, and cannot just do the same actions (such as going to the kitchen) if it is not their restaurant. Similarly, having a business meeting outside at a restaurant or café involves multiple perspectives as well. When a person is focused on their business meeting partner during their conversation, they have the perspective of a business partner, but when the waiter or the waitress comes to take their order, they temporarily switch to being a guest while ordering, and then afterwards going back to being a business partner. This then raises questions in terms of conditions of switching between perspectives or if multiple perspectives are active at the same time, and they should be merged in a way. One could call this a *temporary perspective on norms*.

A final avenue for future work with regards to the formalisms and the agent deliberation architecture that we want to mention here is the integration of goals that require a multi-step planning process. What we mean by this are future goals that the agents can select which are not immediately achieved within one action. For example, we (people) might decide on a Monday that we want to go out to eat on a Friday evening (goal). To achieve this goal we make a plan. The question that arises now is how this plan is affected if the size-based restriction norm becomes active on Tuesday for example. The norm might consider us to replan and reserve a spot or chose a different day to go out to eat or it might not affect us at all. Furthermore, what happens if we meet our friends 'by accident' during the week. It might be enough to cancel the restaurant visit, as our belonging need to socialize with them is satisfied now, or not. These are all questions to consider, as pointed out in Chapter 3.

The second are of future work is with regards to the **support of non-expert users**, as presented in Chapter 5. We provided only the start and the first steps on a long path of norm-focused interaction tools. Therefore, we call for a new research agenda that has these kinds of tools in its focus. This is to support the non-expert user (in our case policymakers) in the process of the abstract analysis of potential effects of norms and their consequences. The interaction tools that we proposed are only the first step and need to be implemented and used in future focus group studies to understand if this is indeed what the users need and what further tools need to be developed and which improvements have to be made to the existing tools that we proposed in this thesis. The research by Schimpf & Castellani [134] is a promising start for focusing on the cognitive aspects and dimensions. Furthermore, new technology developments in artificial intelligence, such as large language models (LLMs), can support the user in adding norms to the system, by for example identifying similar norms, or in the analysis of the behavior of the agents, by for example identifying trends, see here our discussion in Section 5.4.1 in Chapter 5.

6.4 Concluding Remarks

In this thesis, we set out to increase social realism with a focus on norms in social simulations to better support policymakers in their decision-making process. To achieve this, we did not only advance the way in which agents can deal with norms based on what they mean for them, namely through their perspective and the underlying agent deliberation architecture, but also increasing the usability of the simulations by preliminary research on a novel interaction tool.

Moreover our research shows that for social simulations to fulfill their potential of being a powerful tool for policymakers and other decision-makers, closer collaboration between them (policymakers and other decision-makers) and social simulation modelers (us) is needed. This is not only to increase the usability of the simulation, but also to increase our understanding of the policymaking world. This is important, because we (modelers) can then better understand what our agent deliberation architecture need to provide to be used by policymakers and what elements need to be present in the architecture such that they can be used in an interaction tool for policymakers and other decision-makers to interact with the simulation.

Finally, closer collaboration and communication allows for a more fruitful collaboration and can highlight to policymakers and other decision-makers the benefits of integrating social simulations in their decision-making process.

References

[1] Ahrweiler, Petra, Frank, Demian, and Gilbert, Nigel. “Co-designing social simulation models for policy advise: lessons learned from the INFSO-SKIN study”. In: *2019 Spring Simulation Conference (SpringSim)*. IEEE. 2019, pp. 1–12.

[2] Ajzen, Icek. “The theory of planned behavior”. In: *Organizational behavior and human decision processes* 50.2 (1991), pp. 179–211.

[3] Anderson, Elizabeth. “Dewey’s Moral Philosophy”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Zalta, Edward N. Winter 2019. Metaphysics Research Lab, Stanford University, 2019.

[4] Andrews, Benjamin R. “Habit”. In: *The American Journal of Psychology* 14.2 (1903), pp. 121–149.

[5] Andrighetto, Giulia et al. “On the immergence of norms: a normative agent architecture”. In: *In Proceedings of AAAI Symposium, Social and Organizational Aspects of Intelligence, Washington DC*. Citeseer. 2007.

[6] Balci, Osman. “Verification, validation, and testing”. In: *Handbook of simulation* 10.8 (1998), pp. 335–393.

[7] Ball, Philip. “Social science goes virtual”. In: *Nature* 448.7154 (Aug. 2007), pp. 647–648. ISSN: 1476-4687. DOI: 10.1038/448647a. URL: https://doi.org/10.1038/448647a.

[8] Belfrage, Michael, Lorig, Fabian, and Davidsson, Paul. “Simulating Change: a Systematic Literature Review of Agent-Based Models for Policy-Making”. In: *2024 Annual Modeling and Simulation Conference (ANNSIM)*. 2024, pp. 1–13. DOI: 10.23919/ANNSIM61499.2024.10732569.

[9] Bicchieri, Cristina. *The grammar of society: The nature and dynamics of social norms*. Cambridge University Press, 2005.

[10] Boella, Guido and Der Torre, Leendert van. “Constitutive norms in the design of normative multiagent systems”. In: *International Workshop on Computational Logic in Multi-Agent Systems*. Springer. 2005, pp. 303–319.

[11] Boshuijzen - van Burken, Christine et al. “Agent-based modelling of values: The case of value sensitive design for refugee logistics”. In: *JASSS: Journal of Artificial Societies and Social Simulation* 23.4 (2020).

[12] Brennan, Geoffrey et al. *Explaining norms*. Oxford University Press, 2013.

[13] Brenner, Thomas and Werker, Claudia. "Policy advice derived from simulation models". In: *Journal of Artificial Societies and Social Simulation* 12.4 (2009), p. 2.

[14] Broekens, Joost et al. "Do you get it? User-evaluated explainable BDI agents". In: *German Conference on Multiagent System Technologies*. Springer. 2010, pp. 28–39.

[15] Broersen, Jan et al. "The BOID architecture: conflicts between beliefs, obligations, intentions and desires". In: *Proceedings of the fifth international conference on Autonomous agents*. Montreal, Quebec, Canada: Association for Computing Machinery, 2001, pp. 9–16.

[16] Brown, Calum et al. "Behavioral models of climate change adaptation and mitigation in land-based sectors". In: *Wiley Interdisciplinary Reviews: Climate Change* 8.2 (2017), e448. DOI: `https://doi.org/10.1002/wcc.448`.

[17] Butler, Shane. "Child protection or professional self-preservation by the baby nurses? Public health nurses and child protection in Ireland". In: *Social Science & Medicine* 43.3 (1996), pp. 303–314.

[18] Campennì, Marco et al. "Normal= Normative? The role of intelligent agents in norm innovation". In: *Mind & Society* 8.2 (2009), pp. 153–172.

[19] Cariani, Fabrizio et al., eds. *Deontic Logic and Normative Systems: 12th International Conference, DEON 2014, Ghent, Belgium, July 12-15, 2014. Proceedings*. Springer Cham, 2014.

[20] Castelfranchi, Cristiano et al. "Deliberative Normative Agents: Principles and Architecture". In: *Intelligent Agents VI LNAI 1757*. Ed. by Jennings, Nicholas R. and Lespérance, Yves. Springer, 2000, pp. 364–378.

[21] Cialdini, Robert B. and Trost, Melanie R. "Social Influence: Social Norms, Conformity and Compliance". In: *The Handbook of Social Psychology*. Ed. by Gilbert, D. T., Fiske, S. T., and Lindzey, G. McGraw-Hill, 1998, pp. 151–192.

[22] Conte, Rosaria, Andrighetto, Giulia, and Campennì, Marco, eds. *Minding norms: Mechanisms and dynamics of social order in agent societies*. Oxford University Press, 2014.

[23] Conte, Rosaria and Castelfranchi, Cristiano. *Cognitive and social action*. Garland Science, 1995.

[24] Cranefield, Stephen et al. "No Pizza for You: Value-based Plan Selection in BDI Agents." In: *IJCAI*. 2017, pp. 178–184.

[25] Crawford, Sue ES and Ostrom, Elinor. "A grammar of institutions". In: *American political science review* (1995), pp. 582–600.

[26] Dastani, Mehdi, Torroni, Paolo, and Yorke-Smith, Neil. "Monitoring norms: A multi-disciplinary perspective". In: *The Knowledge Engineering Review* 33 (2018), pp. 1–22.

[27] David, Nuno, Fachada, Nuno, and Rosa, Agostinho C. “Verifying and validating simulations”. In: *Simulating Social Complexity: A Handbook* (2017), pp. 173–204.

[28] Davidsson, Paul. “Agent based social simulation: A computer science view”. In: *Journal of artificial societies and social simulation* 5.1 (2002).

[29] De Bruin, Anique BH and Gog, Tamara van. “Improving self-monitoring and self-regulation: From cognitive psychology to the classroom”. In: *Learning and Instruction* 22.4 (2012), pp. 245–252.

[30] Debreu, Gerard et al. “Representation of a preference ordering by a numerical function”. In: *Decision processes* 3 (1954), pp. 159–165.

[31] Dechesne, Francien et al. “No smoking here: values, norms and culture in multi-agent systems”. In: *Artificial Intelligence and Law* 21.1 (2013), pp. 79–107.

[32] Di Tosto, Gennaro and Dignum, Frank. “Simulating social behaviour implementing agents endowed with values and drives”. In: *Multi-Agent-Based Simulation XIII*. Springer, 2012, pp. 1–12.

[33] Dignum, Frank. “Autonomous agents and social norms”. In: *ICMAS’96 Workshop on Norms, Obligations and Conventions*. Citeseer. 1996, pp. 56–71.

[34] Dignum, Frank. “Autonomous agents with norms”. In: *Artificial Intelligence and law* 7 (1999), pp. 69–79.

[35] Dignum, Frank. “Interactions as social practices: towards a formalization”. In: *arXiv preprint arXiv:1809.08751* (2018).

[36] Dignum, Frank. “Foundations of Social Simulations for Crisis Situations”. In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 15–37. ISBN: 978-3-030-76397-8.

[37] Dignum, Frank, ed. *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-76397-8.

[38] Dignum, Frank. “Should we make predictions based on social simulations?” In: *International Journal of Social Research Methodology* 26.2 (2023), pp. 193–206. DOI: 10.1080/13645579.2022.2137925.

[39] Dignum, Frank, Kinny, David, and Sonenberg, Liz. “From desires, obligations and norms to goals”. In: *Cognitive science quarterly* 2.3-4 (2002), pp. 407–430.

[40] Dörner, Dietrich et al. “A simulation of cognitive and emotional effects of overcrowding”. In: *Proceedings of the Seventh International Conference on Cognitive Modeling*. Edizioni Goliardiche Triest, Italy. 2006, pp. 92–98.

[41] Duggleby, Wendy. “What about focus group interaction data?” In: *Qualitative health research* 15.6 (2005), pp. 832–840.

[42] Epstein, Joshua M. “Why model?” In: *Journal of artificial societies and social simulation* 11.4 (2008), p. 12.

[43] Epstein, Joshua M. *Generative social science: Studies in agent-based computational modeling*. Princeton University Press, 2012.

[44] Étienne, Michel. *Companion modelling: a participatory approach to support sustainable development*. Springer, 2013.

[45] Fagundes, Moser Silva et al. “Design and evaluation of norm-aware agents based on Normative Markov Decision Processes”. In: *International Journal of Approximate Reasoning* 78 (2016), pp. 33–61.

[46] Ferretti, Luca et al. “Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing”. In: *science* 368.6491 (2020), eabb6936.

[47] Frantz, Christopher et al. “nADICO: A nested grammar of institutions”. In: *PRIMA 2013: Principles and Practice of Multi-Agent Systems: 16th International Conference, Dunedin, New Zealand, December 1-6, 2013. Proceedings 16*. Springer. 2013, pp. 429–436.

[48] Frantz, Christopher K and Siddiki, Saba. “Institutional Grammar 2.0: A specification for encoding and analyzing institutional design”. In: *Public Administration* 99.2 (2021), pp. 222–247.

[49] Fudenberg, Drew, Levine, David K, et al. “Learning with recency bias”. In: *Proceedings of the National Academy of Sciences* 111 (2014), pp. 10826–10829.

[50] Galán, José M. et al. “Checking Simulations: Detecting and Avoiding Errors and Artefacts”. In: *Simulating Social Complexity: A Handbook*. Ed. by Edmonds, Bruce and Meyer, Ruth. Cham: Springer International Publishing, 2017, pp. 119–140.

[51] Gatto, Joseph et al. “Text Encoders Lack Knowledge: Leveraging Generative LLMs for Domain-Specific Semantic Textual Similarity”. In: *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*. Ed. by Gehrmann, Sebastian et al. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 277–288. URL: `https://aclanthology.org/2023.gem-1.23/`.

[52] Ghorbani, Amineh, Bruin, Bart de, and Kreulen, Kurt. “Studying the Influence of Culture on the Effective Management of the COVID-19 Crisis”. In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 189–230.

[53] Ghorbani, Amineh et al. “MAIA: a framework for developing agent-based social simulations”. In: *JASSS-The Journal of Artificial Societies and Social Simulation* 16.2 (2013), p. 9.

[54] Gibson, Eleanor J and Walker, Arlene S. “Development of knowledge of visual-tactual affordances of substance”. In: *Child development* (1984), pp. 453–460.

[55] Gilbert, Nigel and Troitzsch, Klaus. *Simulation for the social scientist*. McGraw-Hill Education (UK), 2005.

[56] Gilbert, Nigel et al. "Computational modelling of public policy: Reflections on practice". In: *Journal of Artificial Societies and Social Simulation* 21.1 (2018).

[57] Goble, Lou and Meyer, John-Jules Ch. *Deontic Logic and Artificial Normative Systems: 8th International Workshop on Deontic Logic in Computer Science, DEON 2006, Utrecht, The Netherlands, July 12-14, 2006, Proceedings*. Vol. 4048. Springer, 2006.

[58] Graham, Sonia et al. "The social values at risk from sea-level rise". In: *Environmental Impact Assessment Review* 41 (2013), pp. 45–52.

[59] Grignard, Arnaud and Drogoul, Alexis. "Agent-based visualization: a real-time visualization tool applied both to data and simulation outputs". In: *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[60] Groeneveld, J. et al. "Theoretical foundations of human decision-making in agent-based land use models – A review". In: *Environmental Modelling & Software* 87 (2017), pp. 39–48. DOI: https://doi.org/10.1016/j.envsoft.2016.10.008.

[61] Grossmann, Igor et al. "Folk standards of sound judgment: Rationality Versus Reasonableness". In: *Science Advances* 6.2 (2020), eaaz0289. DOI: 10.1126/sciadv.aaz0289. URL: https://www.science.org/doi/abs/10.1126/sciadv.aaz0289.

[62] Habermas, Jürgen. *The theory of communicative action: Volume 1: Reason and the rationalization of society*. Vol. 1. Beacon press, 1985.

[63] Hansson, Sven Ove. "Deontic diversity". In: *Deontic Logic and Normative Systems: 12th International Conference, DEON 2014, Ghent, Belgium, July 12-15, 2014. Proceedings 12*. Springer. 2014, pp. 5–18.

[64] Harbers, Maaike, Meyer, John-Jules, and Bosch, Karel van den. "Explaining simulations through self explaining agents". In: *Journal of Artificial Societies and social simulation* 12.3 (2009), p. 6.

[65] Heidari, Samaneh. "Agents with Social Norms and Values: A framework for agent based social simulations with social norms and personal values". PhD thesis. Utrecht University, 2022.

[66] Heidari, Samaneh, Jensen, Maarten, and Dignum, Frank. "Simulations with values". In: *Advances in Social Simulation: Looking in the Mirror. Looking in the Mirror*. Cham: Springer, 2020, pp. 201–215.

[67] Heine, Bernd-O, Meyer, Matthias, and Strangfeld, Oliver. "Stylised facts and the contribution of simulation to the economic analysis of budgeting". In: *Journal of Artificial Societies and Social Simulation* 8.4 (2005).

[68] Hinch, Robert et al. "Effective configurations of a digital contact tracing app: A report to NHSX". In: *en. In:(Apr. 2020). Available here. url:* `https://github.com/BDI-pathogens/covid-19_instant_tracing/blob/master/Report` (2020).

[69] Hofstede, Geert and Hofstede, Gert J. "Cultural dimensions". In: *www.geert-hofstede. com* (2003).

[70] Hofstede, Gert Jan, Jonker, Catholijn M, and Verwaart, Tim. "Cultural differentiation of negotiating agents". In: *Group Decision and Negotiation* 21 (2012), pp. 79–98.

[71] Hofstede, Gert Jan et al. "Gender differences: the role of nature, nurture, social identity and self-organization". In: *Multi-Agent-Based Simulation XV: International Workshop, MABS 2014, Paris, France, May 5-6, 2014, Revised Selected Papers*. Springer. 2015, pp. 72–87.

[72] Houthakker, Hendrik S. "Revealed preference and the utility function". In: *Economica* 17.66 (1950), pp. 159–174.

[73] Hurk, Mijke van den. "The Effectiveness of Closing Schools and Working at Home During the COVID-19 Crisis". In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 121–138.

[74] Hurk, Mijke van den, Dechesne, Mark, and Dignum, Frank. "No Numbers: Qualitative Structural Validation of Explanatory Social Agent-Based Models". In: *Presented at Social Simulation Conference 2024 in Cracow, Poland*. 2024, pp. 1–14.

[75] Jensen, Maarten, Vanhée, Loïs, and Kammler, Christian. "Social simulations for crises: from theories to implementation". In: *Social simulation for a crisis: Results and lessons from simulating the COVID-19 crisis*. Ed. by Dignum, Frank. Springer, 2021, pp. 39–84.

[76] Jensen, Maarten et al. "Deployment and Effects of an App for Tracking and Tracing Contacts During the COVID-19 Crisis". In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 167–188.

[77] Jones, Andrew J. I. and Sergot, Marek. "On the characterization of law and computer systems: the normative systems perspective". In: *Deontic Logic in Computer Science: Normative System Specification*. USA: John Wiley & Sons, Inc., 1994, pp. 275–307. ISBN: 0471937436.

[78] Kammler, Christian, Dignum, Frank, and Wijermans, Nanda. "Utilizing the Full Potential of Norms for the Agent's Decision Process". In: *Advances in Social Simulation*. Ed. by Squazzoni, Flaminio. Cham: Springer Nature Switzerland, 2023, pp. 193–205.

[79] Kammler, Christian, Dignum, Frank, and Wijermans, Nanda. "Towards a Social Simulation Interaction Tool for Policy Makers—A New Research Agenda to Enable Usage of More Complex Social Simulations". In: *Advances in Social Simulation*. Ed. by Elsenbroich, Corinna and Verhagen, Harko. Cham: Springer Nature Switzerland, 2024, pp. 163–176.

[80] Kammler, Christian and Mellema, René. "Testing and Adaptive Testing During the COVID-19 Crisis". In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Springer, 2021, pp. 139–166.

[81] Kammler, Christian, Mellema, René, and Dignum, Frank. "Agents Dealing with Norms and Regulations". In: *Multi-Agent-Based Simulation XXIII*. Ed. by Lorig, Fabian and Norling, Emma. Cham: Springer International Publishing, 2023, pp. 134–146.

[82] Kammler, Christian et al. "Changing Perspectives: Adaptable Interpretations of Norms for Agents". In: *Multi-Agent-Based Simulation XXII*. Ed. by Van Dam, Koen H. and Verstaevel, Nicolas. Cham: Springer, 2022, pp. 139–152.

[83] Kaur, Harjot and Sharma, Aastha. "Sanction Enforcement for Norm Violation in Multi-agent Systems: A Cafe Case Study". In: *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology: ICDECT 2017*. Springer. 2019, pp. 325–335.

[84] Kebede, Bereket. "Rational Choice Theory". In: *Encyclopedia of Quality of Life and Well-Being Research*. Ed. by Michalos, Alex C. Dordrecht: Springer Netherlands, 2014, pp. 5401–5403.

[85] Keil, Frank C. "Explanation and understanding". In: *Annu. Rev. Psychol.* 57.1 (2006), pp. 227–254.

[86] Kleijnen, Jack PC. "Validation of simulation, with and without real data". In: 1998-22 (1998). URL: https://ssrn.com/abstract=138010.

[87] Klügl, Franziska. "A validation methodology for agent-based simulations". In: *Proceedings of the 2008 ACM symposium on Applied computing*. 2008, pp. 39–43.

[88] Kreulen, Kurt et al. "How culture influences individual behavior during a pandemic: a social simulation of the COVID-19 crisis". In: *JASSS: Journal of Artificial Societies and Social Simulation* 25.3 (2022).

[89] Levy, Andrea Gurmankin et al. "Misrepresentation and nonadherence regarding COVID-19 public health measures". In: *JAMA network open* 5.10 (2022), pp. 1–14.

[90] Li, Tingting et al. "Contextualized institutions in virtual organizations". In: *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*. Springer. 2013, pp. 136–154.

[91] Lorig, Fabian et al. "Comparative Validation of Simulation Models for the COVID-19 Crisis". In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Springer, 2021, pp. 331–352.

[92] Mahmoud, Moamin A et al. “A review of norms and normative multiagent systems”. In: *The Scientific World Journal* 2014.1 (2014), p. 684587.

[93] Malle, Bertram F. “How people explain behavior: A new theoretical framework”. In: *Personality and social psychology review* 3.1 (1999), pp. 23–48.

[94] Maslow, AH. “A theory of human motivation”. In: *Psychological Review google schola* 2 (1943), pp. 21–28.

[95] Mazrekaj, Deni and De Witte, Kristof. “The impact of school closures on learning and mental health of children: Lessons from the COVID-19 pandemic”. In: *Perspectives on Psychological Science* 19.4 (2024), pp. 686–693.

[96] McClelland, David C. *Human Motivation*. Cambridge Univ. Press, 1987.

[97] McShane, Katie. “Why environmental ethics shouldn’t give up on intrinsic value”. In: *The Ethics of the Environment*. Routledge, 2017, pp. 59–77.

[98] Melchior, Alexander. “Economics During the COVID-19 Crisis: Consumer Economics and Basic Supply Chains”. In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 231–268.

[99] Melchior, Alexander, Dignum, Frank, and Yolum, Pinar. “Purposes of and Principles for ABM’s in Policy Development: A Proposal”. In: *Extended Abstract: Social Simulation Conference 2021* (2021).

[100] Melchior, Alexander, Dignum, Frank, and Yolum, Pinar. “Policy Development for Societal Challenges: The Collaborative Agent Based Modelling Policy Framework”. In: *Advances in Social Simulation*. Ed. by Elsenbroich, Corinna and Verhagen, Harko. Cham: Springer Nature Switzerland, 2024, pp. 235–248.

[101] Mellema, René, Jensen, Maarten, and Dignum, Frank. “Social rules for agent systems”. In: *arXiv preprint arXiv:2004.12797* (2020).

[102] Mellema, René, Jensen, Maarten, and Dignum, Frank. “Social Rules for Agent Systems”. In: *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII*. Ed. by Aler Tubella, Andrea et al. Cham: Springer, 2021, pp. 175–180.

[103] Meneguzzi, Felipe et al. “BDI reasoning with normative considerations”. In: *Engineering Applications of Artificial Intelligence* 43 (2015), pp. 127–146.

[104] Mercuur, Rijk, Dignum, Frank, and Kashima, Yoshihisa. “Changing habits using contextualized decision making”. In: *Advances in social simulation 2015*. Springer. 2017, pp. 267–272.

[105] Meyer, J-J Ch, Meyer, John-Jules Ch, and Hoek, W van der. *Epistemic logic for AI and computer science*. 41. Cambridge University Press, 2004.

[106] Meyer, J-J Ch, Wieringa, Roel J, and Dignum, Franciscus Petrus Maria. “The role of deontic logic in the specification of information systems”. In: *Logics for databases and information systems*. Springer, 1998, pp. 71–115.

[107] Meyer, John-Jules Ch. “A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic.” In: *Notre Dame J. Formal Log.* 29.1 (1988), pp. 109–136.

[108] Meyer, John-Jules Ch. “Deontic dynamic logic: a retrospective”. In: *Filosofiska Notiser* 6.1 (2019), pp. 63–76.

[109] Meyer, John-Jules Ch and Wieringa, Roel J. *Deontic logic in computer science: normative system specification*. John Wiley & Sons, Inc., 1994.

[110] Meyer, John-Jules Ch. and Wieringa, Roel J., eds. *Deontic logic in computer science: normative system specification*. USA: John Wiley & Sons, Inc., 1994. ISBN: 0471937436.

[111] Minkov, Michael and Hofstede, Geert. “The evolution of Hofstede’s doctrine”. In: *Cross cultural management: An international journal* 18.1 (2011), pp. 10–20.

[112] Morgan, David L. *Focus groups as qualitative research*. Sage publications, 1988.

[113] Negahban, Ashkan and Yilmaz, Levent. “Agent-based simulation applications in marketing research: an integrated review”. In: *Journal of Simulation* 8 (2014), pp. 129–142.

[114] Nelson, Thomas O. “Consciousness and metacognition.” In: *American psychologist* 51.2 (1996), p. 102.

[115] Nelson, Thomas O. and Narens, Louis. “Metamemory: A theoretical framework and new findings”. In: *Psychology of learning and motivation*. Vol. 26. Elsevier, 1990, pp. 125–173.

[116] Nishizaki, Ichiro, Katagiri, Hideki, and Oyama, Toshihisa. “Simulation analysis using multi-agent systems for social norms”. In: *Computational Economics* 34 (2009), pp. 37–65.

[117] North, Michael J. et al. “Complex adaptive systems modeling with Repast Simphony”. In: *Complex Adaptive Systems Modeling* 1.1 (2013), p. 3. ISSN: 2194-3206. DOI: `10.1186/2194-3206-1-3`. URL: `https://doi.org/10.1186/2194-3206-1-3`.

[118] Oberkampf, William L and Trucano, Timothy G. “Verification and validation benchmarks”. In: *Nuclear engineering and Design* 238.3 (2008), pp. 716–743.

[119] Onwuegbuzie, Anthony J et al. “A qualitative framework for collecting and analyzing data in focus group research”. In: *International journal of qualitative methods* 8.3 (2009), pp. 1–21.

[120] Panagiotidi, Sofia, Alvarez-Napagao, Sergio, and Vázquez-Salceda Javier, Javier. "Towards the Norm-Aware Agent: Bridging the Gap Between Deontic Specifications and Practical Mechanisms for Norm Monitoring and Norm-Aware Planning". In: *Coordination, Organizations, Institutions, and Norms in Agent Systems IX*. Ed. by Balke, Tina et al. Vol. 8386. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 346–363.

[121] Panagiotidi, Sofia, Vázquez-Salceda, Javier, and Dignum, Frank. "Reasoning over norm compliance via planning". In: *Coordination, Organizations, Institutions, and Norms in Agent Systems VIII*. Ed. by Aldewereld, Huib and Sichman, Jaime Simão. Springer. 2012, pp. 35–52.

[122] Păstrăv, Cezara and Dignum, Frank. "Norms in social simulation: balancing between realism and scalability". In: *Advances in Social Simulation: Looking in the Mirror*. Springer. 2020, pp. 329–342.

[123] Păstrăv, Cezara et al. "Social Simulations for Crises: From Models to Usable Implementations". In: *Social Simulation for a Crisis: Results and Lessons from Simulating the COVID-19 Crisis*. Ed. by Dignum, Frank. Cham: Springer International Publishing, 2021, pp. 85–117. ISBN: 978-3-030-76397-8.

[124] Păstrăv, Cezara Maria. "Social simulation for socio-ecological systems: An agent architecture for simulations of policy effects". PhD thesis. Utrecht University, 2022.

[125] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2020. URL: https://www.R-project.org/.

[126] Ramanath, Ana Maria and Gilbert, Nigel. "The design of participatory agent-based social simulations". In: *Journal of Artificial Societies and Social Simulation* 7.4 (2004).

[127] Ramcilovic-Suominen, Sabaheta and Epstein, Graham. "Towards an analytical framework for forest law compliance". In: *International Forestry Review* 14.3 (2012), pp. 326–336.

[128] Rao, Anand S, Georgeff, Michael P, et al. "BDI agents: From theory to practice." In: *Proceedings of the First International Conference on Multiagent Systems*. Ed. by Lesser, Victor and Gasser, Les. Vol. 95. Cambridge, MA, USA: MIT Press, 1995, pp. 312–319.

[129] Rohan, Meg J. "A rose by any name? The values construct". In: *Personality and social psychology review* 4.3 (2000), pp. 255–277.

[130] Rokeach, Milton. *Understanding human values*. Simon and Schuster, 2008.

[131] Sargent, Robert G. "Verification and validation of simulation models". In: *Proceedings of the 2010 winter simulation conference*. IEEE. 2010, pp. 166–183.

[132] Sargent, Robert G. "Verification and validation of simulation models". In: *Journal of simulation* 7.1 (2013), pp. 12–24.

[133] Savarimuthu, Bastin Tony Roy and Cranefield, Stephen. "A categorization of simulation works on norms". In: *Normative Multi-Agent Systems*. Ed. by Boella, Guido et al. Vol. 9121. Dagstuhl Seminar Proceedings (DagSemProc). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2009, pp. 1–20. DOI: `10.4230/DagSemProc.09121.3`.

[134] Schimpf, Corey and Castellani, Brian. "Approachable modeling and smart methods: a new methods field of study". In: *International Journal of Social Research Methodology* (2022), pp. 1–15.

[135] Schlaufer, Caroline et al. "The narrative policy framework: a traveler's guide to policy stories". In: *Politische Vierteljahresschrift* 63.2 (2022), pp. 249–273.

[136] Schnaubert, Lenka. "Providing cognitive and metacognitive awareness information to support regulation in individual and collaborative learning settings". PhD thesis. Dissertation, Duisburg, Essen, Universität Duisburg-Essen, 2019, 2019.

[137] Schwartz, Shalom. "A theory of cultural value orientations: Explication and applications". In: *Comparative sociology* 5.2-3 (2006), pp. 137–182.

[138] Schwartz, Shalom H. "An overview of the Schwartz theory of basic values". In: *Online readings in Psychology and Culture* 2.1 (2012). URL: `https://doi.org/10.9707/2307-0919.1116`.

[139] Schwartz, Shalom H and Bilsky, Wolfgang. "Toward a universal psychological structure of human values." In: *Journal of personality and social psychology* 53.3 (1987), p. 550.

[140] Sen, Sandip and Airiau, Stéphane. "Emergence of norms through social learning." In: *IJCAI*. Vol. 1507. 2007, p. 1512.

[141] Serdeczny, Olivia, Waters, Eleanor, and Chan, Sander. *Non-economic loss and damage in the context of climate change: understanding the challenges*. 3/2016. discussion paper, 2016.

[142] Siddiki, Saba et al. "Dissecting policy designs: An application of the institutional grammar tool". In: *Policy Studies Journal* 39.1 (2011), pp. 79–103.

[143] Simon, Herbert A. "A behavioral model of rational choice". In: *The quarterly journal of economics* (1955), pp. 99–118.

[144] Smajgl, Alex, Izquierdo, Luis R, and Huigen, Marco. "Modeling endogenous rule changes in an institutional context: The adico sequence". In: *Advances in Complex Systems* 11.02 (2008), pp. 199–215.

[145] Squazzoni, Flaminio, Jager, Wander, and Edmonds, Bruce. "Social simulation in the social sciences: A brief overview". In: *Social Science Computer Review* 32.3 (2014), pp. 279–294.

[146] Squazzoni, Flaminio et al. "Computational Models That Matter During a Global Pandemic Outbreak: A Call to Action". In: *Journal of Artificial Societies and Social Simulation* 23.2 (2020), p. 10. ISSN: 1460-7425. DOI: `10.18564/jasss.4298`. URL: `http://jasss.soc.surrey.ac.uk/23/2/10.html`.

[147] Taillandier, Patrick et al. “Building, composing and experimenting complex spatial models with the GAMA platform”. In: *GeoInformatica* 23 (2019), pp. 299–322.

[148] Taillandier, Patrick et al. “Participatory modeling and simulation with the gama platform”. In: *Journal of Artificial Societies and Social Simulation* 22.2 (2019), pp. 1–19.

[149] Van den Bergh, Jeroen CJM, Ferrer-i-Carbonell, Ada, and Munda, Giuseppe. “Alternative models of individual behaviour and implications for environmental policy”. In: *Ecological Economics* 32.1 (2000), pp. 43–61.

[150] Vanhée, Loïs and Dignum, Frank. “Explaining the emerging influence of culture, from individual influences to collective phenomena”. In: *Journal of Artificial Societies and Social Simulation* (2018).

[151] Vanhée, Loïs, Dignum, Frank, and Ferber, Jacques. “Modeling culturally-influenced decisions”. In: *Multi-Agent-Based Simulation XV: International Workshop, MABS 2014, Paris, France, May 5-6, 2014, Revised Selected Papers*. Springer. 2015, pp. 55–71.

[152] Vasconcelos, Wamberto, Kollingbaum, Martin J, and Norman, Timothy J. “Resolving conflict and inconsistency in norm-regulated virtual organizations”. In: *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. 2007, pp. 1–8.

[153] Vaughn, Sharon, Schumm, Jeanne Shay, and Sinagub, Jane M. *Focus group interviews in education and psychology*. Sage, 1996.

[154] Vázquez-Salceda, Javier et al. “From Human Regulations to Regulated Software Agents’ Behavior: Connecting the Abstract Declarative Norms with the Concrete Operational Implementation. A Position Paper”. en. In: *Artificial Intelligence and Law* 16.1 (2008), pp. 73–87. ISSN: 0924-8463, 1572-8382.

[155] Villatoro Segura, Daniel. “Social Norms for Self-policing Multiagent Systems and Virtual Societies”. PhD thesis. Universitat Autònoma de Barcelona, 2011.

[156] Von Wright, Georg Henrik. “Deontic logic”. In: *Mind* 60.237 (1951), pp. 1–15.

[157] Weide, Thomas L van der. “Arguing to motivate decisions”. PhD thesis. Utrecht University, 2011.

[158] Wickham, Hadley. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: `https://ggplot2.tidyverse.org`.

[159] Wieringa, Roel J and Meyer, John-Jules Ch. “Applications of deontic logic in computer science: A concise overview”. In: *Deontic logic in computer science* (1993), pp. 17–40.

[160] Wijermans, Ferdinanda E. H. “Understanding crowd behaviour: simulating situated individuals”. PhD thesis. University of Groningen, 2011.

[161] Wijermans, Nanda et al. "Agent decision-making: The Elephant in the Room-Enabling the justification of decision model fit in social-ecological models". In: *Environmental Modelling & Software* 170 (2023), p. 105850.

[162] Wilensky, Uri. *NetLogo*. http://ccl.northwestern.edu/netlogo/. Northwestern University, Evanston, IL: Center for Connected Learning and Computer-Based Modeling, 1999. URL: `http://ccl.northwestern.edu/netlogo/`.

[163] Wilkinson, Sue. "Focus group research". In: *Qualitative research: Theory, method, and practice* 2 (2004), pp. 177–199.

[164] Winikoff, Michael et al. "Why bad coffee? Explaining BDI agent behaviour with valuings". In: *Artificial Intelligence* 300 (2021), p. 103554.

[165] Wood, Wendy and Neal, David T. "A new look at habits and the habit-goal interface." In: *Psychological review* 114.4 (2007), p. 843.

Appendix A

Apendix: Focus Group Study Scenario and Tasks

Scenario

You are the newly elected major of an average town in Europe. People go to work in the morning by car and come home in the evening. In this town, the following two **social expectations (SE)** hold up:

SE1: A person must be in time for a meeting at the office.

SE2: Employees socialize with their co-workers over a beer and food in a restaurant on Friday after work.

Furthermore, the town is governed by the following **rules (R)**:

R1: A driver must stop their car at a red light.

R2: People are not allowed to drink alcohol when they drive their car.

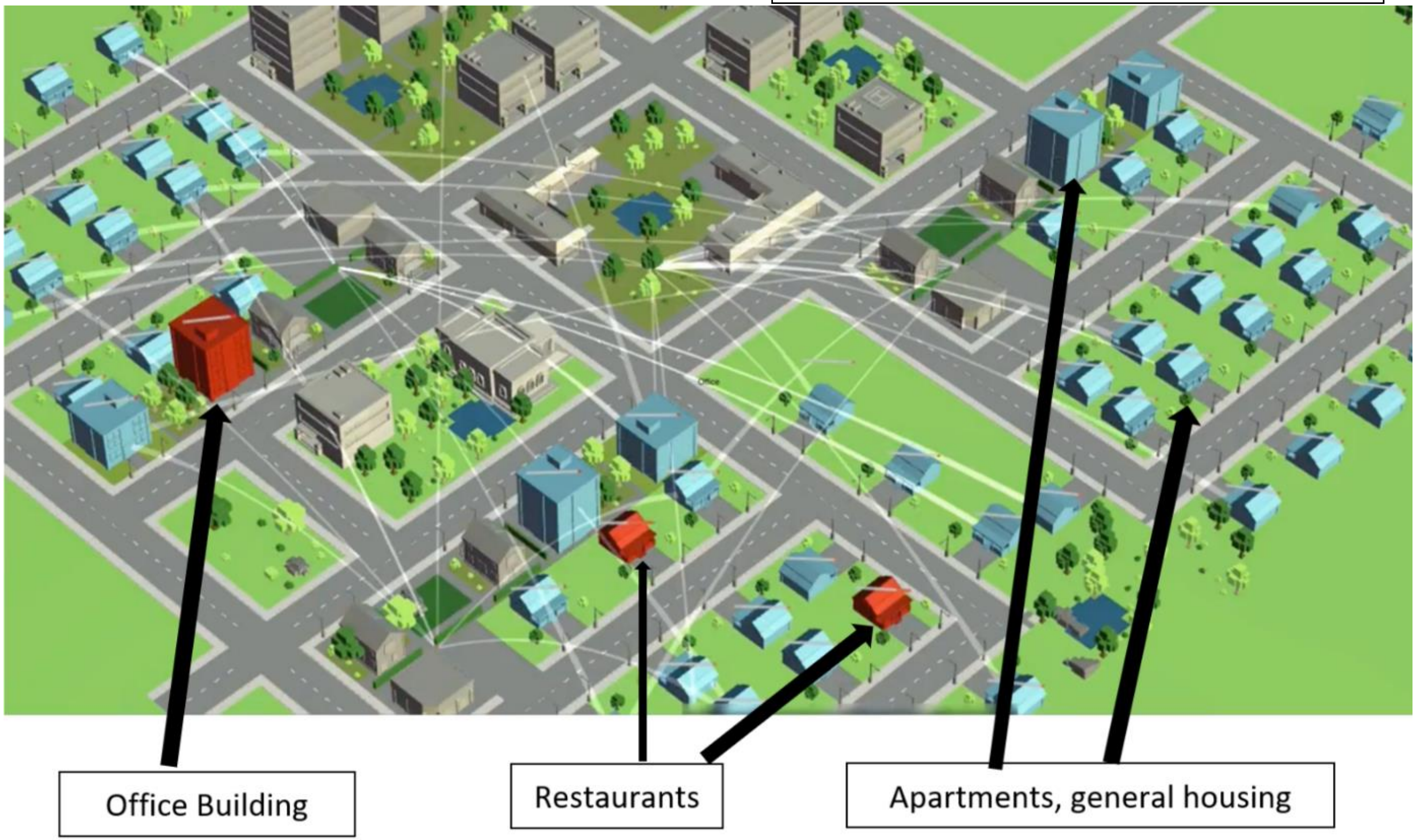
R3: A driver must adhere to the speed limit when driving their car.

The main problem of this town, which you want to solve, is that there are many instances of drunk driving. People go to the restaurant on Friday evening but then drive home in their cars.

To solve this problem and test the effect of potential solutions, the municipality is providing you with a social simulation platform. With this platform you can simulate the potential effects of measures to solve the drunk driving problem.

If you are not sure about the social simulation platform and what it is, you can think about the SimCity or similar city builder games.

Map is not up to scale, people living in housing on the edge are only in driving distance to the office building.



Question 1 – Gathering Information

How would you go about gathering information and understanding the situation at hand?

Examples:

- *Would it be helpful for you to have textual information in form of lists of rules or agents with filter options or order options, such as by context or by most violated*
- *Would "onClick" functions for these lists help to more textual information about the rule (i. e. its details) or the agent (i.e. its current goal, needs satisfaction, …)?*
- *Would it help to see which goals are chosen more often and which actions are mostly taken?*
- *Would graphical information be useful for you, such as the numbers previously mentioned to see their development over time or the amount of people at each restaurant or something else?*
- *Would you like this information be always there for you or would you prefer this information not visible for you and you make it visible once you need it?*

Further Development of the Scenario

After assessing the situation, you get a list of suggestions from your advising committee on potential new policies that you can chose to implement to fight the drunk driving problem, these **suggested interventions (SI)** are:

SI1: Place a police officer in front of every restaurant in the evening to make sure that people who drink do not drive.

SI2: Demand early closing hours of restaurants so people can use public transport.

SI3: You install special shuttle buses at specific times and people must take those buses.

SI4: Taxies / Uber paid by the municipality

SI5: Designated Drivers get free non-alcoholic beverages

To test and see the effects of the suggested interventions, you now chose one of them or your own idea (if you have one) and want to implement it in the social simulation system.

Question 2 – Adding a new Rule

How could the social simulation system assist you with adding the new rule? How would you go about adding it to system?

Examples:

- *Would it help to see in a textual form which other rules the targeted agent group has to comply and are active in the same context?*
- *Or would you prefer all rules that are active in the same context?*
- *Would some building blocks help or lists where you can chose items from for the different parts of the rule or do you prefer free text input?*
- *Would it help if the simulation system tells you that the current formulation of the rule is in conflict with other rules, such as pop warnings or other forms?*
- *Would trial simulations (previews) showing how the current rules work and where it is affecting the agent's deliberation process, i. e. where and how the rule is considered by them (if it all)?*
- *Would you like this information always available to you or would you prefer that this information is hidden and can requested by on demand, through for example button clicks or menu interactions?*

Question 3 – Unexpected Consequences

After the implementation of the desired policy, you observe that the effects are not as intended and expected. You observe the following: [Based on your choosing, I will present them to you now]

How would you go about finding out why these effects occur? How can the social simulation system assist you with that?

Examples:

- *Would textual about the goal / action chosen most often by the agents help with a comparison before and after the new/changed rule got active?*
- *Would intersections (when a new/changed rule got active) in the graphs mentioned in the previous question help to get more insights?*
- *Is there other information that you would like to focus on or single out? If so, which information and in what form would you like that information to be?*

Consider now the situation below. A person went to a restaurant and had a couple of drinks with their colleagues to celebrate the promotion of a friend. Because of that, the person is late for a meeting the next morning. A decision must now be made between violating the speed limit and being in time for the meeting or obeying the traffic laws and therefore being late for the meeting.

Question 4 – Identifying a Conflict

How would you like this conflict to be presented to you or in general be shown that there is even a conflict and that the agent must make a decision here?

Examples:

- *Would you like a textual information with some highlight (colors or other) to show that the highlighted rules are conflict or that the agent violated a rule?*
- *Would you like the simulation to automatically pause and show a pop-up warning saying that there is a rule conflict and then highlighting the agent and the situation?*
- *Would you like a trial simulation (preview) of the simulation in isolation where the agent makes all possible decisions and then it is highlighted which rule is violated for each time?*

Question 5 – Resolving the Conflict

How would you like the social simulation system to assist you in resolving the conflict?

Examples:

- *In the previously mentioned preview, would you like to have the possibility to set a hard rule for which rule is followed and which one is violated?*
- *Would you like the system to make you (textual) suggestions about which rule(s) could be changed in what way to remove the conflict?*
- *Would you like the social simulation system to show in a previews or textual form, which agent behavior the least amount of conflicts, thereby also showing that some conflicts are not avoidable?*
- *Would you like the social simulation system to tell you in a pop-up warning if there is no conflict free solution for the current conflict, given the current set of rules, and that further rule changes/additions/removals are necessary?*

Question 6 – Generalizing from the given Situation

In general, how would you like the information of conflicting situations and rule violations be organized and presented to you?

Examples:

- *Would textual groupings of the rules by agent and/or activity context with colored highlight of conflicting rules be helpful to you?*
- *Would you like to have trial simulations (previews) of all currently existing conflicts available to you that you can chose from a list and see?*
- *Would you like to have a network graph of rules where rules are connected if they have the same object, activity context, or agent?*
- *Such a graph could then have some "onClick" functions to further inspect the rule and a specific connection?*
- *If such a graph would be helpful, would it help even more if conflict rules would have their connections colored differently or any other form of highlight to identify the conflict faster?*
- *Would you like to have any combination of the information possibilities mentioned so far where you can turn on or off parts of them?*
- *If so, how would you like this combination to look like?*