An Integrated Process Mining and Data Mining Approach for the Validation of Agent-based Simulation Models

Author
Ruben R. Govers

UT Supervisors
Dr. Faiza Bukhsch
Dr. Amin Asadi
Rob Bemthuis MSc.

A thesis presented for the degree of
Master Business and Information Technology

EEMCS
University of Twente
The Netherlands
21st of July, 2023

Abstract

Agent-based simulations and process mining are upcoming fields. However, literature research found that research on agent-based models regularly lacks proper validation or an understanding of this validation. Moreover, the bounds of input parameters in which a model operates validly are often unknown either. Uncertainty regarding this might cause an agent-based model to become invalid as it gets adjusted over time. This thesis presents an innovative approach for validating agent-based simulation models based on a three-cycle interpretation of CRISP-DM, a renowned data mining methodology. Each cycle aligns with a separate discipline. The first involves agent-based simulation, the second process mining, and the third data mining. The thesis follows the design science research methodology to deliver the approach. As part of this methodology, four objectives were identified. The approach should be interdisciplinary, adaptable, effective, and interpretable. The approach was demonstrated and evaluated through a detailed case study of the Schelling model of segregation. The input parameters that influence the model in the case study include topology, heterogeneity, and agent rules. The case study shows that the approach achieved the objectives within the research context. The findings highlight the potential of this integrated approach across a range of fields, creating a new path for further advancements in the field of agent-based models. Examples of such advancements include automatic agent-based model generation from event logs or the creation of interpretable agent-based models using data mining.

CONTENTS

1 INTRODUCTION

This introduction section aims to explain the scope and goals of the research. First, a problem statement is provided. Subsequently, a set of research questions is provided.

1.1 Introduction

This research focuses on an approach for validating agent-based models using process mining. Agent-based models (ABM) are widely used in modeling and simulation. There is no single definition, as the applicability of agent-based models is wide. They are generally defined as an approach that models self-organizing systems, both social and technical, composed of autonomous and interacting agents [45]. An agent is an autonomous entity capable of decision-making, as explored in Subsection 2.1. Examples of theoretical and practical applications for agent-based models are optimization problems, sociological systems, simulating agri-food supply chains, and more [5, 10, 78].

Agent-based modeling involves creating a representation of the real world. An extension of this is agent-based simulation (ABS), which allows the collection of data or looking into agent interactions. The data can then be used for optimized business decision-making and data generation for machine learning [37]. There exists a wide range of techniques with which ABM research and development can be conducted and combined. Some examples are combinations of simulations, process mining, business process models, ontologies, workflows, human interactive agents, and robot process automation [3, 29, 31, 33, 35, 43, 49, 73].

Promising for this research is the field of process mining (PM), an upcoming area of research that is extensively described by van der Aalst [79]. Process mining is a technique that, unlike traditional data mining techniques, focuses on end-to-end processes based on an event log. Process mining is traditionally used to discover, analyze and improve business processes [81]. However, research has shown that it can be used for analyzing agent-based models too. Agent-based simulations have been used to generate data for PM purposes so that the business process model does not need to be developed by hand [72]. Process mining has also been used to assess how changes in an agent-based model affect the model [20, 27]. Furthermore, process mining has been used in combination with agent-based models for validation purposes [51], which is also what this research aims to do.

Another field of interest is data mining (DM). Data mining was born from the increasing prevalence of larger databases [30]. The data has to be turned into information before decisions can be made, which is where data mining techniques come to use. Data can be clustered, classified, or used for prediction in order to make informed business decisions. The combination of data mining and process mining requires more research, as it is presented in 2 out of 11 challenges in the process mining manifesto [81]. It has been used in the context of educational process mining as well as the analysis of underground mining (the type that extracts rocks rather than information) [11, 13]. Agent-based models and data mining are commonly combined, for example in order to investigate agent-based simulation results, analyze social simulations, or better predict crime [4, 21, 52, 62]. It seems that agent-based models, and especially agent-based simulation models, combined with process mining and data mining are still separated despite the potential for collaboration between these fields. Through this combination, one can enhance the validation process by incorporating empirical data, improving the model's accuracy, and gaining a more comprehensive understanding of the modeled system. The integration that the approach in this research provides can lead to more robust and reliable agent-based simulation models, as well as better insights into the underlying processes and mechanisms they represent.

1.2 Problem Statement

In order to gain usable information, one has to capture a model of reality to a high degree in an agent-based model. Verification techniques are used for technical correctness, to assert that the model is bug-free. This work focuses on the validity of the agent-based models, which is at least as important if not more so. Validation is concerned with the degree to which the model is an accurate representation of a real-world system it is intended to simulate [65]. In the research topics structured literature review, it was found that 18 out of 24 papers in the quality appraisal had a poor score regarding the validation of the method. This shows that validation, despite the fact that there are a variety of tools available, is still a rather neglected part of agent-based modeling. Another issue is that changing real-life situations might mean that agent-based simulation models in use become invalid: new input parameter combinations can lead to unexpected emergent behavior. Furthermore, the interpretability of agent-based models, simulations, and process model is often low due to low-level behavior having high-level effects and complex visualizations.

Thus, there is a need for a validation approach that solves these problems of validity, maintainability, and interpretability. This work does so through an approach for agent-based simulation model validation using process mining and data mining. This type of validation is reusable for adjusted agent-based models (i.e., ABMs with a different set of input parameters), for example when a business hires extra people and wants to adjust their simulation by changing a parameter. Using the proposed approach, one can gain insight into the functioning of the agent-based model as well as how it is classified as valid or invalid. This approach is based on the CRISP-DM methodology [67] which provides a strong foundation. The aim is to explore a new direction of ABM validation research and show that both PM and DM can be viable components of an agent-based simulation model validation approach. The research methodology is developed according to the DSRM by Peffers et al [54]. According to the methodology, the approach is demonstrated and validated through a case study on the Schelling model [66].

1.3 Research Questions

The first research goal is to solve the challenges related to agent-based simulation model validity and by extension maintainability and interpretability. The second goal is to combine the fields of agent-based models, process mining, and data mining and show its potential. The main question is presented as follows:

"How can an agent-based simulation model's validity be assessed using process mining and data mining techniques in the context of the model's topology, heterogeneity, and agent behavior rules?"

A set of additional research questions is presented with the goal of answering the main question effectively. These are as follows:

1. What are the key factors and definitions when assessing the validity of an agent-based simulation model in the context of the main research question?
2. What are suitable methodologies for the development of a validation approach for agent-based simulation models?
3. How can this methodology be implemented for the validation approach?
4. How can this implementation for the validation approach be operationalized?
5. Does the proposed validation approach for agent-based simulation models solve the research problems?

1.4 Structure

This section covered a brief introduction to the subjects relevant to this thesis, the research problem, and the research questions. The following section will explore background information that will help answer the first research question in Section 2. The sections after this correspond to the research methodology. This research methodology and the methodology on which the approach is based are explained in Section 3. Subsequently, Section 4 covers the proposed approach and case study setup. Section 5 contains the execution of the performed case study. The case study results and their implications for the proposed approach are treated in the discussion, in Section 6. Finally, the research questions are revisited and directions for future research are proposed in Section 7. An overview of the structure aligned with the DSRM methodology can be found in Figure 5 of Section 3.

2 BACKGROUND

The goal of this background section is to introduce the concepts, definitions, and scope of ABM, ABS, PM, and DM relevant to this research. Several general use cases and tools used in these fields are highlighted. After this, the topic of validity is briefly explored. Finally, the results of the structured literature review are provided in order to underline the scientific relevance of this research.

2.1 Agent-based Models and Simulations

2.1.1 Definitions

This subsection of the background section aims to briefly explore ABM, ABS, and the concepts relevant to research sub-question a. As stated in the introduction, there is no single definition of what constitutes an agent-based model. Abdou et al [1] define agent-based modeling as "a computational method that enables a researcher to create, analyze, and experiment with models composed of agents that interact within an environment". Here, agents are considered distinct parts of a program that represent social actors. Another definition, according to Macal and North [45], is that agent-based modeling is a new approach to modeling systems comprised of autonomous, interacting agents. They state that a single definition for an agent is also non-existent, but that it can be assumed that agents are autonomous, self-contained, and social. Furthermore, an agent might live in an environment, pursue a goal, learn and adapt, and have resource attributes. This work assumes the second definition.

Agent-based models in the context of this research are used for agent-based simulation models. Simulation requires some clarification, as it is associated with a number of fields that yield their own definitions. Simulations were generally performed through discrete-event simulation, differential equations, and system dynamics [70]. Shannon states that agent-based simulation is the process of designing an agent-based model of a real system and conducting experiments with this model [69]. The purpose of these experiments is to understand the behavior of the system and evaluate various strategies for the operation of the system. This work follows that definition. Some attributes that help us understand and discern ABS from previous simulation techniques are:

1. an individual-based bottom-up approach as opposed to a process-oriented top-down modeling approach;
2. a decentralized thread of control, as agents are autonomous, rather than a centralized approach;
3. active entities with a representation of intelligence rather than passive entities;
4. no concept of queues;
5. no concept of flows and modeled macro behavior, which is emergent from micro decisions in agent-based models; and
6. input distributions that are often based on theories or subjective data rather than collected or measured data.

2.1.2 Tools and use cases

The previous subsection defined ABM and ABS in this work. The following paragraph first aims to show some applications of agent-based simulations in order to develop a better understanding. Following this, tools for implementation are shown.

A set of use cases is discussed first. Molina et al [46] used agent-based modeling and simulation for air traffic management. The goal was to model new collaborative decision processes for flow traffic management, which it did successfully. Agent-based simulation models proved to be practical for use at an intermediate level of abstraction. Carley et al [15] developed a scalable citywide multi-agent network numerical model, called BioWar. BioWar simulates individuals as agents who are embedded in social, health, and professional networks and tracks the incidence of background and maliciously introduced diseases. Another practical use case was the use of Covasim in the Covid-19 pandemic by Kerr et al [39]. The researchers developed an agent-based model in Python to model the pandemic and possible intervention strategies that were utilized for policy-making across the globe. Valid models are essential for these use cases since lives are at stake in these domains.

Several tools for implementing agent-based models and simulations will be listed in this paragraph. The first tool is MAREA: it is covered extensively in the literature and was prevalent in the structured literature review [27, 28, 72, 84]. It is a software application with simulation possibilities, which can be used to present the trading behavior of a company for decision support among others. A commercial tool is Anylogic [75], though most of the tools out there are academic. Examples of these are MASON[44], Repast Suite [18], GAMA [74] and NetLogo [77]. Repast and MESA [57] both have Python support, which makes them accessible. This paper makes use of the Agent4py framework [24] for the case study, as it integrates the many different tasks of agent-based modeling within a single, simple environment. This makes agent-based model development fast and easy, and possible to integrate with other techniques such as PM and DM later on.

2.1.3 Included Concepts

As seen in the primary research question, the problem context involves three concepts. These are topology, heterogeneity, and agent rules. First, a brief explanation of these concepts is provided. We then explore some other concepts that exist but are not explored further in this research.

First, heterogeneity is reviewed. Heterogeneity is defined as “the fact of consisting of parts or things that are very different from each other” by the Cambridge Dictionary [14]. In the context of agent-based model research, it can refer to the heterogeneity of the agent population or to the heterogeneity of the topology. We focus on the heterogeneity of the population. A homogeneous agent population would consist of almost identical agents with the same agent rule. A heterogeneous agent population would consist of agents where the agent rules can differ between individual agents.

Secondly, topology is looked into. Topology in the research of Bohlmann et al refers to “the structure of how individuals (..) are connected” [12]. Macal et al [45] explored agent-based modeling and its concepts, and describe different forms of topology in agent-based models as a way to connect agents. An essential idea, according to their research, is that agents only interact with a limited number of agents within the agent population. In the case study of this research, a static network in the shape of a grid and awareness of direct neighbors is used to adhere to these definitions.

The third and last concept that is relevant to the experiments in this research is the agent behavioral rule, or simply agent rule. This concept describes how sustainable patterns emerge in systems that are completely described by simple, deterministic rules based on only local information [45]. An example of this is how the complex patterns of a flock of birds can be simulated through three simple agent rules that only require information about the local environment, rather than the whole flock [60].

These three concepts are chosen due to their significant influence on the emergent behavior of the system. Heterogeneity captures the diversity amongst agents and relates to the requirement that agents are autonomous. The topology, or environment, is crucial in how agent interactions and decision-making occur. This makes topology an important variable, which is also mentioned in the agent-based model definition by Abdou et al [1]. The last concept, behavioral rules, guides agent actions directly and changes in micro behavior directly result in a change in macro behavior. This makes it an important factor to include. The next subsection briefly describes some excluded concepts, which are adaptation and learning, cooperation, authorization and authentication, goals, and attributes. These are all important factors that deserve to be mentioned. However, they are not prioritized because these concepts would require a complex agent-based model, even though their effect on the emergent behavior is less direct and the scope of the research is focused on validation.

2.1.4 Excluded Concepts

This subsection covers a set of other concepts that could be relevant to this work. These concepts are briefly explained, as they are not operationalized in the experiments but still bear relevance for future research.

First, we look into individual agent adaptation and learning, which can be relevant. For this concept, an agent may have the ability to learn and adapt its behaviors based on experiences that are stored in some kind of dynamic attribute [45]. In other words, this creates a more complex agent with evolving rules.

Doran et al consider cooperation to be a key term in differentiating the field of multi-agent systems from related disciplines, but the definitions surrounding cooperation and communication are “at best unclear and at worst highly inconsistent” [34]. Including experiments on validity surrounding the concepts of communication, cooperation, and coordination might therefore introduce ambiguity.

Furthermore authentication and authorization, which are most relevant in the security domain, are explored by Cremonini et al [19]. The concepts have importance in combination with coordination and topology.

Fourthly, goals and resource attributes are closely related key concepts [45]. Goals can be used to drive an agent's behavior and can possibly give the agent a benchmark to modify its behavior. Resource attributes can be used by an agent to achieve those goals. Examples are the agents' social network, their wealth, energy levels, or even the information they have about their direct environment.

Fifthly, time can be used to alter how the model functions. Different agent actions can have different durations based on statistical distributions, set times, or time steps unrelated to real-world time. It is an important concept for valid models, but experimenting with this requires complex models and analysis related to the real world. It is therefore left out of scope.

Finally, it can be said that this list of concepts is not exhaustive. Grimm et al [26] extensively describe design concepts, most of which align with this and the previous subsections. Additionally, the ODD protocol provided in their paper is a valuable tool for implementing an individual- or agent-based model with these concepts in mind.

2.2 Process Mining

This section aims to briefly highlight what PM entails. PM is a data-driven technique that aims to extract valuable insights and knowledge from event logs. It is used to discover, analyze to improve business processes by process understanding, conformance checking, and process enhancement according to the process mining manifesto [81]. The manifesto also states that process mining explicitly involves real processes from the real world.

As briefly stated in the introduction, event logs serve as the primary input for process mining. Event logs contain information on activities, the associated case or agent, and timestamps [79]. They can also contain used resources and other attributes related to process execution for more elaborate analysis. Sources of event logs can include real-world applications such as an enterprise resource planner, business process management system, or a simulation as done in this thesis. The latter is especially promising since real-world event logs are generally scarce, especially for research purposes. Event logs are often stored in either a CSV format or the XES standard, which is a popular information-sharing protocol for the interchange of event data between information systems in many domains [32].

These event logs are subsequently utilized by a discovery algorithm. These algorithms automatically derive a process model from event logs. Notable event logs include the alpha miner, alpha plus miner, heuristics miner, and inductive miner. However, this list is not exhaustive. The alpha miner is well-known and widely used and works by analyzing the causal relationships between activities in a given event log [80]. The alpha plus miner extends the previous algorithm by including more data from the event log and being able to discover simple loops [2]. The heuristics miner employs heuristics to discover process models based on observed behavior [85]. The inductive miner aims to discover a representative model with a low complexity based on generalizing behavior in the event logs [41].

From the discovery algorithms follows a process model, often in the form of a business process model and notation (BPMN) or a Petri net. Petri nets are considered in this work. These are graphical models that provide a visual representation of a process [55]. They describe a graph-like structure and consist of places, transitions, and arcs. Arcs can be considered to be directed edges that connect a place and a transition or vice versa, but never two transitions or two places. Places and transitions are similar vertices but are disjoint finite sets. Petri nets offer a clear and intuitive way of modeling and analyzing the behavior of processes, where analyzing is often done with the help of so-called tokens. An example of a Petri net can be seen in Figure 1.

The second step in process mining is analysis [81]. Commonly used techniques are conformance checking, performance analysis, social network analysis, and predictive analytics. Especially conformance checking and performance analysis are of use in this thesis. The first entails a comparison between the actual behavior observed in the event logs against the modeled process to identify deviations, bottlenecks, or non-compliance. The performance analysis looks into performance metrics related to processes, such as throughput, cycle time, response time, and resource utilization.

There are several existing process mining tools available that facilitate the discovery and analysis of insights from event logs. ProM, a popular academic PM framework, offers a wide range of functionalities [82]. ProM provides a comprehensive set of plugins for process discovery, conformance checking, and performance analysis. Disco, another prominent but commercial tool by Fluxicon, focuses on user-friendly process mining with features like automated discovery, flexible visualizations, and interactive filtering [23]. Pm4py, which is also featured in this work's case study, is a Python library specifically designed for

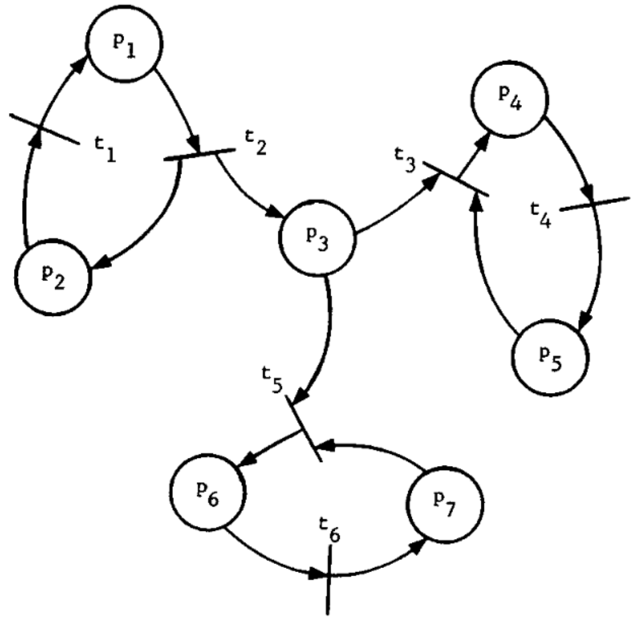

Figure 1. A simple Petri net as described in Peterson [55].

PM tasks. It offers a versatile set of algorithms and methods for data pre-processing, process discovery, and evaluation [56]. Since the tool is a Python library, it is easy to integrate with other tools. Other notable process mining tools include Celonis and RapidMiner [16, 58].

2.3 Data Mining

This section aims to briefly discuss what data mining is, with a focus on classification due to its relevance for this research. Similarly to PM, DM is a data-driven technique for knowledge discovery. Data mining encompasses a wide range of methodologies, including statistical analysis, machine learning, pattern recognition, and predictive modeling, all aimed at extracting valuable information from raw data [30].

Additionally, data mining encompasses the task of classification. Classification involves building predictive models to assign labels to data points. Classification algorithms such as decision trees, support vector machines, and neural networks are commonly used to learn from historical data and make predictions on unseen data. This predictive modeling aspect of data mining enables the creation of models that can assist in tasks across a wide range of domains such as analyzing social systems in agent-based simulations [52], fraud detection [47], and predicting the intensity of workouts [25, 48]. In the case of this work, it assists with the validation of ABS models. The case study in this work relies on decision trees for the classification of ABS models.

Decision trees are a data mining technique used for the classification of labels [71]. The technique classifies a population into segments that resemble branches, which together construct an inverted tree with a root node, internal nodes, and leaf nodes. The root node and internal nodes represent choices. Leaf nodes, or end nodes, represent the final result. All nodes are connected by branches, through which a path is followed based on the decisions made in the nodes. Decision trees generalize on both small data sets and large data sets. Generally, data is split into a training, validation, and test set which is preceded by feature selection (i.e. selecting variables) and data cleaning. Different algorithms that implement a decision tree exist. CART, C4. 5, CHAID, and QUEST are the most popular. They differ in statistical measures, pruning techniques, types of data, and how nodes are split. The main concept of decision trees is the same across these algorithms.

2.4 Validity

This section discusses what is meant by validity, as the term is coined in the research question. It is important to distinguish validation from verification, which is often defined as "ensuring that the computer program of the computerized model and its implementation are correct" [65]. Validation is the degree to which the model is an accurate representation of a real-world system that it is intended to simulate. Validity consists of multiple dimensions, where different subjective techniques (e.g., expert opinion or visuals) or objective techniques (e.g., statistical tests) can be used to evaluate a dimension.

Moreover, let us take a brief look into some other definitions for validation. Barlas [6] makes a distinction between two types of models in order to define validity. He states models can be "causal-descriptive" (theory-like, white-box) or "correlational" (data-driven, black-box). Correlational models. such as forecasting models, can often be validated using the previously stated definition. White-box models require not only quantitative validation but also an explanation of the model behavior. This implies that the interpretability of the validation approach and model is an important objective for this thesis. Moreover, it adds a philosophical component to the definition of validity for each model. It is important to define the validity of each model separately to address this, like the example for this thesis below.

Additionally, validity can also be defined according to its impact on decision-making. One can argue that a model is valid when this model leads to identical decisions as perfect knowledge of the real world. In other words, validity could refer to the usefulness rather than the accuracy of a model. No definition is considered more correct than the other, but this thesis adheres to the first as it aims for accuracy by developing a mostly empirical approach, rather than usefulness which is more difficult to measure as such.

This study allows the model developer to define their own validity criteria, potentially based on a metric that establishes a link between the model and the real world. The link to the real world is crucial for discerning validation from conceptual validation or verification. This is illustrated in Figure 2. In order to provide an example, a definition is made which can be assumed for the case study on the Schelling model in Section 5:

"With respect to the real world, there exists a position for almost every agent where the ratio of similar neighbors to different neighbors is such that the agent is happy according to its similarity preference."

Determining the threshold for "almost every agent" should be based on relevant real-world statistics. A consequence of this requirement is that adhering models eventually reach a state where the number of unhappy agents and agent moves are minimized, and self-termination is achieved or can eventually be achieved. On the other hand, models that do not adhere to this requirement fail to converge and remain in a non-terminating steady state. The approach that this research proposes can be categorized as objective and predictive validity. This is because the classification models, which will be further elaborated in Subsection 5.6.2, predict a state of "will likely converge" and "will likely not converge" based on objective metrics and a reality-based threshold.

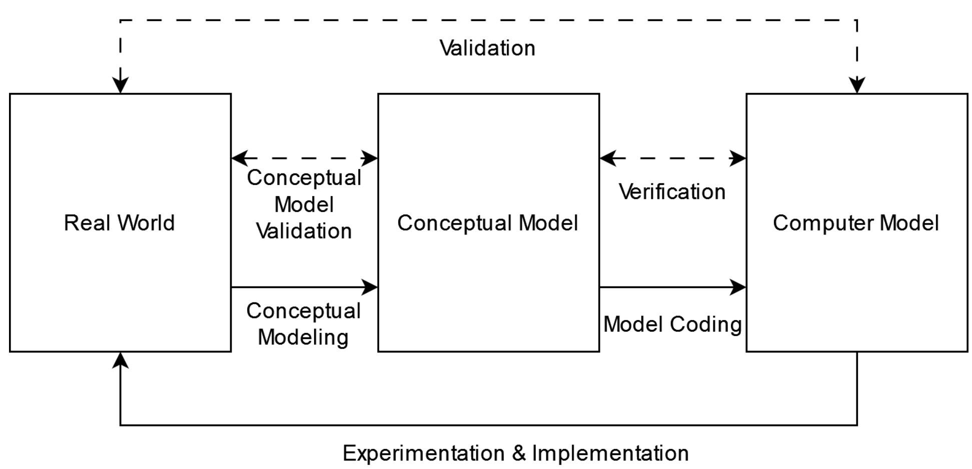

Figure 2. Simulation model verification and validation in the modeling process inspired by Robinson [61].

2.5 Related Research

A structured literature research was performed on identifying research challenges. This work consisted of 359 papers in the original query, with 42 papers evaluated in detail. The study was done in the scope of ABM, PM, and BPM. This section aims to distill information and show research challenges that would elevate the quality of future research in these three fields. The proposed research show why there is a place for this work, which combines ABS, PM, and DM for empirical validation.

The first challenge is that authors often argue for the extension of models and an extension of the validation of these models. Therefore, the first proposed research challenge is that extended validation studies are to be performed by the authors. Important, however, is that this validation is done before the expansion of these models to show both the industry and the scientific community that more research in this area is likely to pay off. The lack of proper validation is in part due to the industry's secrecy and in part due to the varying nature of agent-based artifacts. An attempt to standardize metrics that measure the quality of an agent-based model, in a similar fashion to how machine learning models are trained would be welcome. A proposed approach for this is by combining the fields of ABS and PM.

The second research challenge is the standardization of methodologies that cover models, programming languages, and tools. Despite researchers discussing examples or prototypes of artifacts for the sake of validation or illustration, there is often a lack of documentation for future use or integration with other tools. One example is S-BPM, which according to Fleischmann et al [22] requires to be a complete specification language (despite the extensive research already existing surrounding S-BPM). Standardization and better usability might also increase accessibility for researchers and businesses to use tools for agent-based approaches.

Thirdly, there exists a need for research on reactive and asynchronous agent relations, as opposed to successive and deterministic relations and communication. Numerous simulation studies assume successive relationships between agents that act in an undisturbed environment. Reality does not really match with this- communication noise and unpredictable events might cause a simulation to be overly stable and predictable, or a mismatch with reality. To add to this, it would be interesting to know how different levels of information sharing between agents affect an agent-based model.

This brings us to the next research challenge, which is that knowledge of the impact of a complex versus a simple multiple-agent environment on the quality of a simulation is useful. Noise or unpredictable events in experiments might assist with designing more stable agent-based models for real-world applications.

Furthermore, research on expanding compositional approaches for architecture-aware process models would be useful. The idea that architecture-aware process models can be (re)generated and formally proven to be sound by design is valuable. It does require an expansion of proven interfaces that represent agent interactions and a way to distill these from event logs efficiently.

Visualizations and explanations of an agent-based model are a sixth area of research. Visualizations, including statistics that can enhance business decisions, might help people using such systems to better grasp how the targeted agent-based model functions and how agents interact within it. Examples of beneficiaries are managers attempting to improve business processes in a large retail store or manufacturing plant operators that are able to gain better real-time insight by live-process mining and visualization.

Self-interested and evolving agents have been briefly mentioned in a small number of papers. Research into the effect of more complex agents on the quality of agent-based models and simulations might prove useful for future implementations. This is because not only business environments but also people, roles, and processes within this environment change.

Moreover, only one paper briefly links process mining to both Industry 4.0 and digital twins was found in the SLR. Increased volumes of data and the emergence of self-organizing systems might provide a new use case for process mining and improved production. Though Osman and Ghiran [50] did perform a recent exploratory research in 2019 and did show that combining these fields yields potential benefits, much is to be discovered and it is therefore likely an interesting area to explore. Additionally, data mining is rarely combined with ABM and PM and could prove to be beneficial.

Finally, a research challenge concerning another emerging technology is proposed. Although cloud-based and decentralized data management solutions are relatively common at the time of writing, there might still be a place for agent-based data management solutions. Sandita and Popirlan [64] proposed a multi-agent data management system in JADE. An expansion of this research might lead to interesting results.

The following list summarizes the research challenges covered in this literature review:

- increased validation and using metrics for a stronger foundation of the literature is required;
- standardization of languages and tools;
- research into reactive and asynchronous agent relations over successive and deterministic agent relations;
- expansion of compositional approaches for sound models;
- visualizations and explainable results;
- research into self-interested and evolving agents;
- combining the field of agent-based process mining, data mining, and industry 4.0 concepts;
- and finally extension of research into agent-based data management solutions.

Additionally, Table 1 shows a set of research ideas based on the structured literature research.

Research Direction	**Contribution to Science**
Using PM to validate MAS models according to a set of metrics.	Contribute to the validation methods available to MAS researchers.
Combining the fields of PM and BPM with Industry 4.0 for smarter manufacturing.	Some foundational research has already been done, but more research would increase both the scientific and business value of PM.
Research into the impact of noisy data and unpredictable events on the quality of MAS and PM results.	Increase the possibility for applications of these techniques to real-world problems.
Research into the impact of complex, reactive, and asynchronous agents as opposed to simple, deterministic, and successive agents on the functioning of a MAS.	Most studies in the SLR use simplified agents with a large range of limitations and assumptions. Understanding how emergent behavior differs from more complex models would help with modeling real-world scenarios.
Extension of research in compositional approaches for sound business process models.	By creating more formally proven compositions to be used in process models, the soundness and readability of said models can increase.

Table 1. A set of research ideas based on the identified research gaps of this paper.

2.6 Contribution

This section shows the research challenges this paper addresses, based on the previous subsection.

From the list of research challenges resulting from the literature review, this paper is able to address the first and the fifth. It addresses the first by developing an approach for agent-based simulation models based that rely on a varying set of metrics. It addresses the fifth by showing the use of process models and decision trees to assist the validation approach. These tools can also provide a visual and interpretable insight into agent-based models. Additionally, this research partially aligns itself with the first row of Table 1. This shows that this work deals with multiple areas that require additional research in line with the structured literature review.

2.7 Concluding Remarks

This subsection aims to answer the first research question which sounds as follows:
"What are key factors and definitions when assessing the validity of an agent-based simulation model in the context of the main research question?"

The first key factor consists of agent-based simulation, which is the process of designing an agent-based model for a real system and conducting experiments with this model [69]. After this, included concepts and excluded concepts were discussed in Subsection 2.1.3 and the subsequent subsection.

The second key factor is process mining, a data-driven technique that aims to extract valuable insights from event logs [81].

The third factor which is key to this work, is data mining with a focus on classification. Data mining is another data-driven technique for knowledge discovery, generally aimed at extracting valuable information from raw data [30].

Lastly, the concept of validity is important. Validity is the degree to which the model is an accurate representation of a real-world system that it is intended to simulate [61, 65].

3 METHODOLOGY

This section explores how the research questions are answered. The methodology of the research consists of an adapted version of the design science research methodology (DSRM) by Peffers et al [54]. The developed approach relies on the cross-industry standard process for data mining (CRISP-DM) as defined by Wirth and Hipp [86]. The approach will be explained in this section, as both methodologies can potentially be used in the research methodology, the artifact, or both. The artifact itself is explained in Section 4. First, some background on the used methodologies is provided. After the considerations are explained and finally the research methodology as applied here is described.

3.1 Background

3.1.1 DSRM

The DSRM presented by Peffers et al [54] is a widely used methodology that incorporates principles, practices, and procedures required to carry out design science research in information systems. Design science research generally entails the development of an artifact. The methodology meets three objectives, which makes it a solid foundation for this research: it is consistent with prior literature, provides a nominal process model that can be adapted, and provides a mental model for presenting and evaluating the design science research.

The methodology contains six steps: problem identification and motivation, the definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. While these steps are in sequential order, it can be the case that research starts at any of these steps. The list below provides a brief summary of what these steps entail.

- Problem identification and motivation require a definition of the specific research problem and justification for the value of a solution.
- The definition of the objectives for a solution requires rational inference of objectives from the problem definition and the knowledge of what is possible and feasible. They can be quantitative or qualitative.
- Design and development involve the creation of the artifact. These can be any construct, model, method, or "new property of technical, social, and/or information resources".
- Demonstration and evaluation range from a simple case study that shows the approach to a formal evaluation. A demonstration requires one or more instances of the problem to be solved. Evaluation requires measurements of how well the artifact supports the solution to the problem. At the end of an evaluation, the researcher can step back to the design and development for an improved iteration.
- Communication requires informing the public on the problem and its importance, the artifact, its utility and novelty, the rigor of the design, and its effectiveness.

3.1.2 Crisp-DM

CRISP-DM is considered the de facto process model for applying data mining projects and is aimed at standardizing data mining processes [67, 86]. The term CRISP-DM was first coined in 1996 by a set of companies that had the goal of creating a standardized data mining methodology, before being released in 2000. Figure 3 describes the 6 processes in CRISP-DM and their relations. The steps are defined as follows by the literature:

1. Business Understanding is an assessment to get an overview of the available and required resources.
2. Data Understanding focuses on collecting data from sources, exploring and describing it as well as checking the data quality.
3. Data Preparation is traditionally described as cleaning the data from noise and outliers, based on a set of inclusion and exclusion criteria dependent on the model.
4. Modeling consists of selecting the modeling technique, building the test case, and the model.
5. Evaluation is done to check the results against defined business objectives or to review the research process in general.

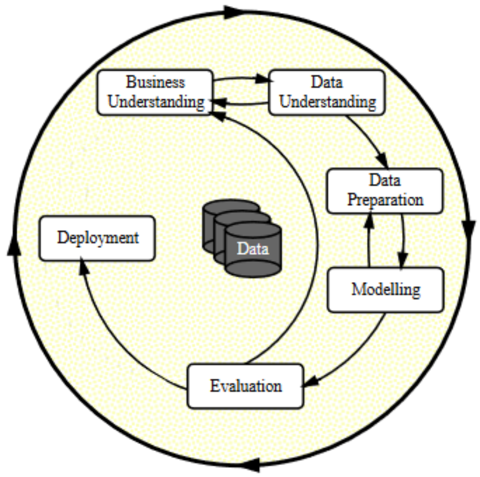

Figure 3. CRISP-DM as described in [86].

6. Deployment describes either a final report or a software component with the project results.

One can observe similarities with DSRM in both the research processes and iterative nature. CRISP-DM was chosen as a foundation for the approach despite its orientation on business rather than research. The advantages of CRISP-DM are that the methodology provides a set of sub-processes for each phase that align closely with the required steps in process mining and data mining. Adapting these steps to steps to align with this research creates a straightforward methodology. The result is explored later in this work.

3.2 Considerations

This subsection discusses the benefits and disadvantages of different configurations that are possible for the research design and solution design. The first consideration discusses the use of DSRM for both processes. The second consideration is the use of CRISP-DM for both processes. The last considerations use CRISP-DM with a nested DSRM and vice versa. The chosen approach is explained in more detail in Section 3.3. Before discussing these configurations, let us briefly look at the differences between the two. DSRM is designed to guide the development of an artifact to solve a design science problem: it is a research methodology. CRISP-DM is a process model specifically structured for data mining problems and it aligns closely with the techniques used in data mining. The key differences can be found in Figure 4.

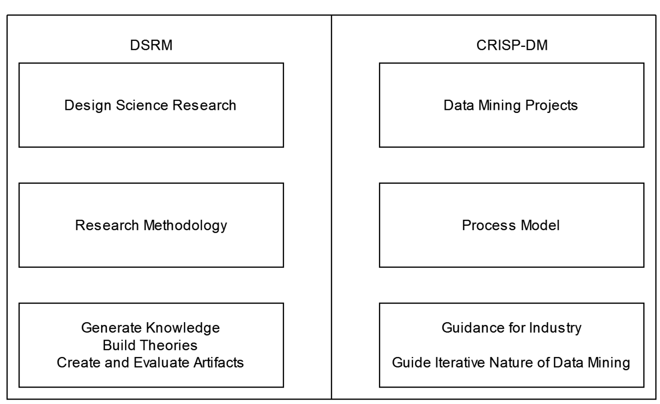

Figure 4. Difference between DSRM and CRISP-DM

By using DSRM as the foundation of the validation approach, a few problems arise. The first is that DSRM is focused on solving a problem with an artifact. Due to the “openness” of this methodology, a structured set of steps regarding the objectives, development, and evaluation of an artifact is missing with respect to process mining and data mining techniques. Secondly, the validation approach is the artifact of this research. This artifact itself does not produce a design science research artifact but a data mining artifact.

Using CRISP-DM for the research methodology has a similar problem. If CRISP-DM were to be used for the research methodology, there would be a gap consisting of the problem investigation, clear objectives, and communication phase. Additionally, CRISP-DM phases with sub-steps in for example "data preparation", are unsuitable for a design science research methodology. After all, how would one normalize and remove duplicates from a background section?

This demonstrates that using DSRM for the validation approach or CRISP-DM for the research methodology is unsuitable and that using either technique for both is likely a bad idea. Using DSRM for the validation approach may limit the ability to leverage the well-established and widely-used techniques provided by CRISP-DM for data mining tasks. Similarly, relying on CRISP-DM may overlook the specific considerations and requirements of design science research. Benefits are highest when using both techniques for their intended purposes: it leads to a more robust research and solid foundation for the validation approach.

3.3 Research Methodology

This subsection of the methodology dives into how DSRM is applied in this research. A schematic can be seen in Figure 5. Each step will be discussed sequentially. The validation approach and how CRISP-DM relates to it are explained in more detail in Section 4.

The first step, the definition of the problem and its motivation, was largely performed in the research topics course in the form of a structured literature review on research gaps.

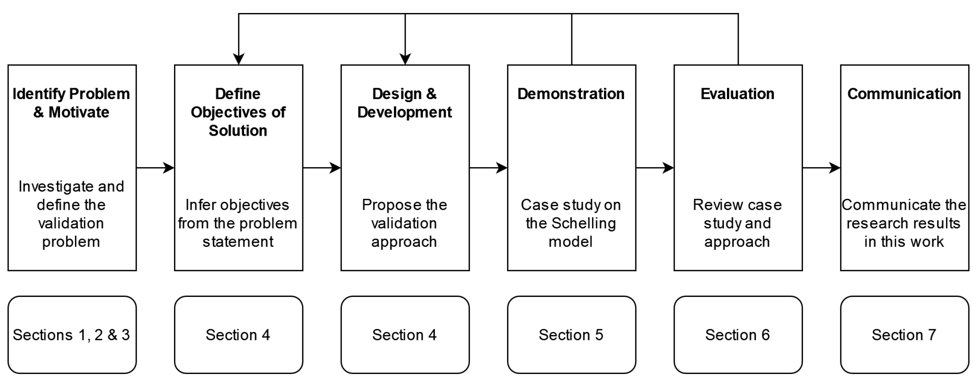

Figure 5. Overview of the research methodology used in this research according to the DSRM process model.

This was followed by the development of research objectives. Each objective was inferred from the problem statement, which was provided in the introduction in Section 1. This phase has been revisited numerous times throughout the research, as the research questions and subsequent objectives required to be tuned to the artifact in development. These objectives are presented in Section 4.

After the objectives were set, the solution was designed and developed. The solution is presented in Section 4. The design and development phase consisted of several brief design cycles, where iterations were made between the design and development phase and the demonstration phase in the shape of a case study. The first step was to determine that CRISP-DM would be a suitable foundation for combining ABS, PM, and DM. After this, the agent-based simulation phase was designed and executed as part of the case study. Following this, the design and development phase was revisited to design the process mining phase before applying it to the case study. Finally, the data mining phase was designed and demonstrated as well. The method was refined one last time before adjusting the case study, after which it was ready for evaluation.

The evaluation can be found in the shape of a discussion in Section 6. The evaluation has two goals. The first goal is to evaluate the case study. The second goal is to see how the case study complies with the objectives of the solution.

The communication phase of the DSRM has the purpose of communicating the results of all previous research stages combined. It corresponds to the conclusion, which can be found in Section 7.

3.4 Concluding Remarks

This subsection answers the second research question:
"What are suitable methodologies for the development of a validation approach for agent-based simulation models?"

Based on the considerations in this section, it was decided that a good methodology for the development is the DSRM. Additionally, it was decided that CRISP-DM is a suitable approach for the validation approach itself.

4 SOLUTION DESIGN

This section describes the methodology of the approach, the research artifact, based on CRISP-DM, which can be found in Figure 7 and is explained in Section 3.

4.1 Objectives of the Approach

First, a set of objectives is designed. According to the DSRM [54], they are to be derived from the problem statement which can be found in Subsection 1.2. The goal is that the research problems are solved if these objectives are completed while answering the main research question. The derived objectives are as follows:

- Interdisciplinary: the approach combines the fields of ABS, PM, and DM successfully.
- Adaptability: the components in the approach are replaceable (e.g., replace one PM algorithm with another).
- Efficacy: the approach is effective in dealing with the validation problem.
- Interpretability: the approach is interpretable.

The first objective refers to the approach being interdisciplinary, not to be confused with multidisciplinary. The latter would assume the relevant disciplines are used in parallel but disjoint, whereas the first assumes the approaches are intertwined. While it is not empirically measurable, logical inference can be used. The second objective, adaptability, means that the cycles in the approach are to be technically disjoint. While information can be shared between them, the techniques should not depend on each other. This allows the approach to be more durable as the research field progresses and to be more customizable for different applications. This objective is not empirically measurable either but can be inferred from both the model in Figure 7 and the case study in Section 5. The third objective, efficacy, suggests the approach is effective with respect to the validation of an agent-based simulation model. This objective can be inferred partially from the metrics in the data mining cycle of the case study as provided in Subsection 5.6. The final objective, interpretability, refers to the ability to understand the agent-based simulation model as well as the decision-making with respect to validation in the data mining cycle. It can therefore be deducted from the case study. The following paragraphs aim to explain the proposed solution with these objectives in mind.

4.2 Considerations

There are two possible interpretations of CRISP-DM that can be considered for the methodology. The first interpretation involves integrating ABS, PM, and DM steps across the original phases described in the CRISP-DM methodology, which is shown as the first interpretation in Figure 6. This resembles a single design cycle from business understanding up to evaluation. While this interpretation may seem appealing, it poses challenges as agent-based simulations and process mining are fields that might not align well with CRISP-DM, and the intertwining of steps complicates the replacement of tools and techniques. Hence, the second interpretation deviates more from the original methodology but aligns better with the research objectives and adaptability.

The proposed approach is performed as a three-cycle approach, where each cycle essentially represents a CRISP-DM iteration. A high-level abstraction is the right approach in Figure 6. The first cycle focuses on gaining an understanding of the model and problem, generating the event logs, and generating additional data needed for the following cycles. These event logs are analyzed and data is extracted using process mining tools in the second cycle. The last cycle uses this extracted data as input for data mining tools to discover whether these can be used for automated validation.

As mentioned previously, CRISP-DM offers a few generic steps per cycle. It was chosen to not adapt each of these steps according to the literature surrounding CRISP-DM. The first reason for this is that the techniques surrounding ABS and PM do not completely align themselves with the steps in data mining, despite some resemblance. The second reason is that the iterations in this research do not exist to improve a single data-mining technique to reach deployment, as is usually the case with CRISP-DM use cases. Rather, each cycle employs its own technique. This means that traditionally very important steps like data cleaning do not have to occur in each iteration.

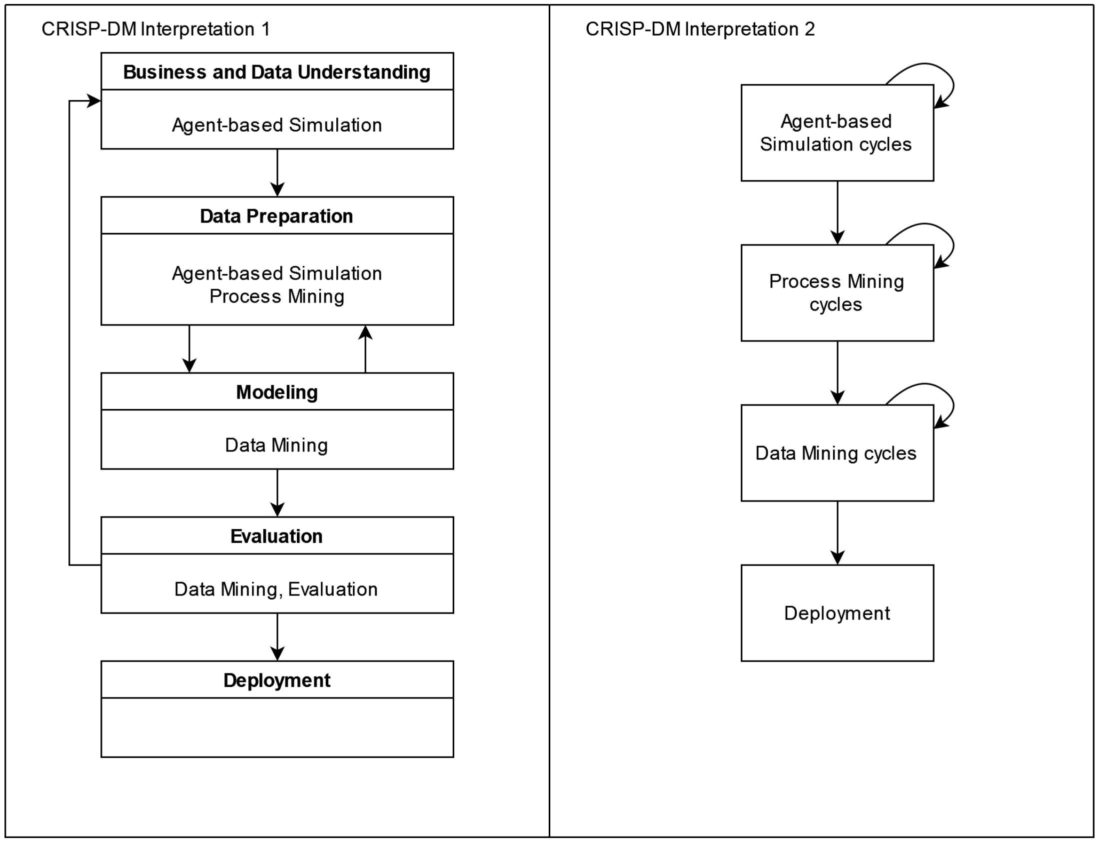

Figure 6. A high-level abstraction of initial approaches.

Finally, we will briefly look at the flow of information between the iterations. An image representation can be seen in Figure 8. The arrows between the iterations, which are derived from Figure 7, indicate the information that is used. The squares indicate all the information generated. For example, the agent-based simulation generates event logs for process mining, and certain portions of the information are utilized to enhance the understanding of the agent-based model. On top of this, the extracted information, specifically the move metric, is employed later in the training of classifiers.

4.3 The Approach

This subsection will examine each of the steps as described in Figure 7. The high-level boxes indicate a CRISP-DM cycle consisting of phases, where each phase contains a set of steps. We start with defining the main goal, then the agent-based model for the ABS cycle, and sequentially follow the steps until the performance evaluation of the data mining cycle. The flow of information between each cycle is covered in the case study below.

4.3.1 Main Goal

First, the main goal for the validation approach is to be defined. This generally concerns itself with the validation of a specific ABS model and its application to the real world, although it can vary depending on which insights one aims to obtain. The purpose of this main goal is to align the intermediate objectives and techniques chosen in the first phase of each cycle.

4.3.2 Agent-based Simulation Cycle

The first step is to choose an agent-based model and determine the objectives of the simulation cycle, with data generation for the following cycles and model understanding kept in mind. This highly depends on the use case, but two arbitrary examples might be to “collect event logs over 20 model configurations” or to “learn how the agent density affects the model output”. One might pick multiple models to compare at the end of the study, design a model, or simply pick an existing model that might be subject to change in the future. Some conditions are that the complexity of the model must be tolerable for PM tools and that

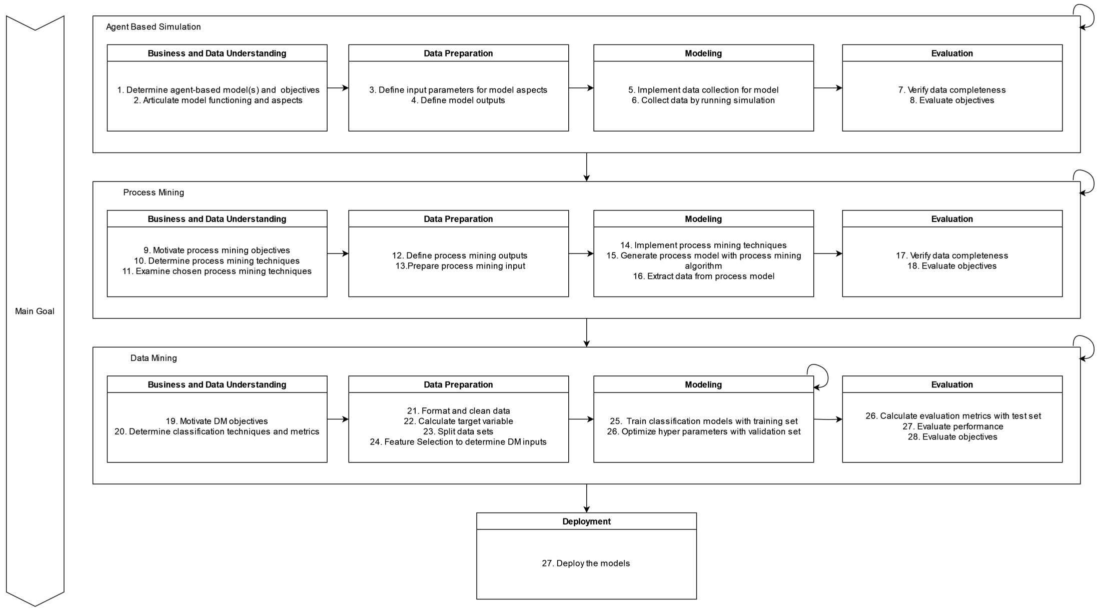

Figure 7. Overview of the approach methodology. The names correspond to the CRISP-DM processes and each overlapping square represents an iteration of the methodology. A larger version is available in Appendix C, Figure 16.

sufficient data must be available for the DM cycle. The second step is to gain an understanding of how the model functions. A better understanding assists with performing the subsequent steps. This understanding can initially be achieved by looking into assumptions in the agent-based model and reasoning about its functioning, as well as literature research. In a possible following iteration of the ABS cycle, one might consider performing sensitivity analysis to get a better understanding of the actual model functioning.

In the data preparation phase, one has to choose input parameters for different aspects of the model. One can do this based on steps 1 and 2, with knowledge of what the model responds to and which aspects are relevant. The output of the model also has to be defined before the modeling steps are done. The latter includes information that is required for the PM and data mining cycles such as event logs, the number of events, and execution length among others.

The modeling phase does not necessarily refer to the implementation of the ABS model itself, as this is not a development but a validation approach. Thus, the next steps are to implement data collection according to the previously defined model output and run a simulation on the chosen parameters to collect data. One should consider running each configuration of the model over multiple iterations with predefined seeds for random-number generators. This allows for more reliable data for analysis, a higher volume of data for the process mining and data mining cycles, and the repeatability of the experiment.

The final phase requires an evaluation of the ABS outcome, which will assist with both the process mining and data mining cycles. This phase consists of a brief check on data completeness. This check includes ensuring the variables capture the final state of the simulation run, and that all simulations within the intended parameter ranges are performed. Finally, one can determine that the objectives are achieved or that another iteration of the ABS cycle is needed.

4.3.3 Process Mining Cycle

Once the agent-based simulation steps are performed, process mining objectives are ready to be determined. An example of this could be "extract a sound Petri net from the simulation event logs". The techniques can be selected based on this. These techniques consist of the tools for implementation, the discovery algorithm for the simulated event logs, the type of process models, and other possible algorithms for distilling information from the resulting process models. It is important to gain an understanding of the relevant techniques in this step, for example by checking the documentation, scientific grounds of the implementations, and required data and data types.

First, the process mining outputs are to be defined based on the techniques that are chosen and the objectives that are set. After this, the output of the ABS cycle is then to be prepared for process mining input. This could mean that data cleaning (e.g., removing outliers and noise) is required. Data also has to be stored and formatted in a way that is accepted by the chosen process mining techniques. Examples of this are ensuring that the date-time format is correct and that the files are stored according to the XES format.

Once the input is prepared and output goals are set, one can implement the chosen techniques with the chosen tools. After this, one can iterate over the ABS-generated event logs and data to extract the process models and other information such as cycle times or the number of transitions in the process model.

Last but not least, there is a brief evaluation of the PM cycle. The goal determine whether the data is complete according to the previously defined outputs, or whether more can be extracted. Additionally, the objectives are to be evaluated. If these objectives are not achieved sufficiently or if incompleteness of the data is determined, one can decide to re-iterate this cycle.

4.3.4 Data Mining Cycle

The first step in the final cycle is to determine the classification objectives, for example, to "classify the test set with at least 80% accuracy". This also involves defining how the target variable is determined. Several suggestions for this are expert opinion or a metric based on model outputs and real-world information. The next step is to choose techniques and metrics. Intuitive metrics and techniques that allow for human interpretability, such as accuracy and decision trees, are preferred. Just like in the previous cycle, it is important to consider the documentation, scientific grounds, and data.

After this, the data collected so far has to be formatted in an input-appropriate manner. One might consider further cleaning the data and normalizing it for real-world data, data prone to outliers, or certain data mining techniques. After this, a validity target variable to be classified has to be calculated (or determined through expert opinion or other means) for the data. Depending on the technique, normalization might also be needed.

The second-last step for data preparation is to split the data into a set of training data, validation data (if one does not apply k-fold cross-validation), and a set of testing data [59]. The validation and test data should consist of parameters that do not occur in the training data, in order to simulate the validation of new real-world situations or models. The goal of the training set is to create a model that can classify the gathered information as valid or not valid. This model can be improved using k-fold cross-validation or the validation set. The test set is reserved until the final evaluation and acts as a simulation of the deployment. It is important that all sets are disjoint such that no parameter setting is to occur in more than one set at the same time.

Following this, the final phase can be applied. This is feature selection. Some DM methods are not sensitive to this step, but it can make a model more lightweight and better performing, and a lower number of features can increase model interpretability [42]. Techniques such as principal component analysis can also help with data understanding and feature selection in this phase [38].

The modeling phase is relatively simple and cyclical in nature. First, the chosen models are trained on the chosen inputs. Either through k-fold cross-validation or a validation set, one can obtain the previously defined metrics. These can be used to tune the model parameters before retraining if wanted. Optionally, one can perform a grid search on a large range of parameters with the goal of finding local optima.

The final steps of the final evaluation phase first require the use of the separately stored test set to calculate the metrics on the final models. The final models are picked according to the metrics calculated in step 20. The metrics can then be used to evaluate the performance of the models in order to pick a final model.

4.3.5 Deployment

Lastly, CRISP-DM also has a phase that represents the deployment. Deployment differs between different users. The main goal is to integrate the validation approach with the target architecture, such that an ABS model can be adopted for its real-world use case.

While this depends on how the approach was applied, it would generally consist of a pipeline between each of the cycles. This pipeline would consist of the following components: simulating the new model, automatically cleaning the output, using that output for process mining, merging the data, and finally automatically obtaining a prediction from the previously trained data mining model by using the futures. One recommendation for implementing such a deployment is that all the tools for each cycle are chosen in such a way that they can be easily replaced, just like how the approach cycles are mostly disjoint.

This is done most easily by using libraries within the same programming language in each cycle, or by connecting unrelated tools through built-in operating system pipelines. Even though it is likely excessive, one can also decide on the use of specialized software for pipelines.

Another recommended technique for the integration of these cycles into a target architecture is the use of integration and implementation concepts. These concepts are available as an extension to the ArchiMate standard and are described by Jonkers et al [36]. Finally, the final classification model will be able to classify new configurations of the ABS model before they are put to use.

4.4 Concluding Remarks

This section aims to answer the third research question. The third research question is as follows:
"How can this methodology be implemented for a validation approach?"

This section found that a three-cycle approach based on CRISP-DM is a solid foundation for implementing such a validation approach, and that is preferable over a single-cycle interpretation.

5 CASE STUDY

This section shows how the approach can be applied to an agent-based simulation model. First, the tools used in the case study are explained. Following the tools, the section is structured according to the cycles, phases, and steps as described in the previous section and Figure 7. For a higher-level overview of the information exchanged between each cycle, we will refer to Figure 8. The evaluation of the case study and approach are provided in the next section.

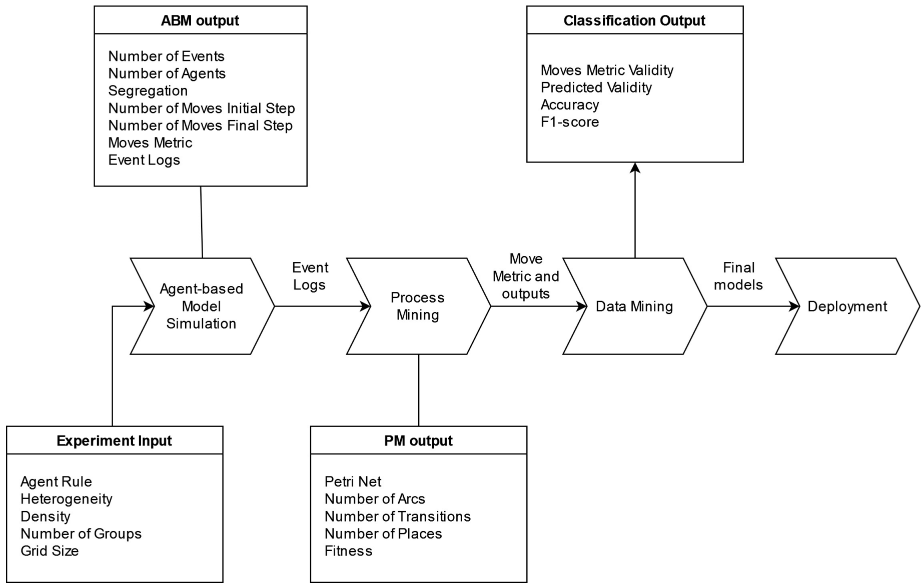

Figure 8. An overview of the information generated and used between the CRISP-DM cycles.

5.1 Tools

It is important to consider the objectives of the approach design when picking the tools. This research is implemented entirely in Python 3.6.9 [83]. It has the benefit that it is easily adapted to a different ABS model. Additionally, it is possible to create a pipeline for the deployment phase in order to automatically validate a new input or easily update the data mining models.

The library used is Agentpy 0.1.5, which provides an object-oriented structure for easily implementing agent-based models [24]. Rather than using ProM or any commercial PM tools, it was decided to use the Pm4py version 2.7.2 [8]. The downside of this Python library is that it lacks user-friendliness, but this comes with the benefit of increased flexibility in terms of creating an experimental framework and increasing the possibilities for future research by extending existing methods. In addition, documentation on its use and possibilities is provided by the Fraunhofer Institute [56], a large German research society. The format used for event logs is the XES standard [32]. This format was chosen because the standard is a generally acknowledged format for the interchange of event data between information systems in many domains. The classifiers used in the experiment use the Scikit-learn 1.0.2 implementation, which is an accessible and popular package for machine learning research [53]. Data processing was done using the Pandas library [76].

5.2 Schelling Model of Segregation

This subsection illustrates the Schelling model on a high level and explains its use case. The case study, which applies the approach to this model, assumes a basic understanding of the segregation model as it only explains the technical details. The segregation model was designed by Schelling to model (racial) segregation [66]. The goal of the model was to illustrate how inadvertent behavior might also be a contributing factor to segregation besides factors such as prejudice, zoning laws, and gentrification. The model showed that agents who had a neutral opinion on being surrounded by agents of a different group still segregated themselves from those agents over time.

Moreover, this model can be considered an agent-based model according to the explored literature in Subsection 2.1. Additionally, it is relatively simple to implement and exhibits interesting macro-behavior with a possible real-world impact on decision-making. However, certain settings in the model lead to a non-terminating steady-state simulation, which will also be our definition of the model being invalid as illustrated in Subsection 2.4 This makes the model a good subject for the case study.

5.3 Main Goal

Before starting with the methodology, the main goal has to be described. In this case, it is to validate the Schelling model according to the definition previously given in Subsection 2.4: "With respect to the real world, there exists a position for almost every agent where the ratio of similar neighbors to different neighbors is such that the agent is happy according to its similarity preference."

5.4 Agent-based Simulation Cycle

5.4.1 Business and Data Understanding

The first step is to set objectives for the ABS cycle, before picking an agent-based model. The objectives are:

- to collect event logs and information about agent-based simulation performance; and
- to gain insight into how segregation between agents is affected by the topology and agent rule.

The agent-based model used for the research is an adaptation of Schelling's model of segregation [66], and its implementation is inspired by Bemthuis and Lazarova-Molnar [7]. The model illustrates how an agent's tendencies regarding their neighboring agents in a grid can lead to a certain degree of segregation. The model topology consists of a square grid. Each cell in this grid contains at most one agent. Multiple agent groups exist. Each agent belongs to one such group and bears a set tolerance towards other groups, also known as the agent rule. This tolerance is checked for each agent in every time step. If the tolerance is not met, an agent moves to a random unoccupied cell. This repeats until all agents are 'satisfied' or until a certain number of steps have been executed.

Various parameters are used to influence experiment outcomes. These parameters include topology, agent heterogeneity, and agent rule complexity as mentioned in the background section. Heterogeneity refers to the diversity of rule sets used (either homogeneous for all groups or distributed per group). Agent rule complexity relates to the agent rule (i.e., the threshold that defines an agent's happiness with regard to their neighbor) and the number of agent groups. Additionally, the topology can be changed in terms of network size and agent density. The network size refers to the size of the two-dimensional grid (e.g. 20 by 20 cells). The agent density refers to the ratio of occupied cells to empty cells. The number of steps in the simulations is capped at 100 steps.

Metric	Values	Number of Steps
Size	50x50 and 20x20	-
Threshold	0.2 - 0.95	10
Number of Groups	1-4	4
Density	0.2 - 0.99	10
Heterogeneity	False	-

Table 2. Initial parameter search ranges for homogeneous experiments

Metric	Values	Number of Steps
Size	20x20	-
Threshold	0.30 - 0.95	5
Number of Groups	2 and 3	-
Density	0.50 - 0.90	5
Heterogeneity	True	-

Table 3. Initial parameter search ranges for heterogeneity experiments

As part of this phase, it is important to describe how the model behaves. In the first iteration after the development of the model, a parameter search was performed. The parameters for this search can be found in Table 2 for experiments with homogeneous agents, i.e., every agent has the same agent rule. Heterogeneous simulations took significantly longer and were therefore performed on a smaller range of parameters, which can be found in Table 3.

Only two graphs relevant to the objective are described. However, more information can be extracted from the parameter search. The graphs can be found in Appendix A, where the size is 20x20, and averages of the averages across the dependent variable are calculated over all other parameter ranges combined. We start with the topology parameters. As can be seen in Figure 13, high densities are likely unsuitable for valid models (according to the definition defined below). It can be observed that the average number of steps and the average number of non-terminating runs increases, whilst the segregation largely remains the same. It can be argued that the number of moves per step is lower as there are only a few cells available, and the probability of having a sufficient amount of similar neighbors per move is low if the agent rule is above 50%. The second component of the topology relates to the grid size. As can be seen in Table 8, the grid size had very little impact on the segregation and number of steps.

In Figure 14, we can observe some other effects. The segregation decreases as the agent rule increases: agents are more prone to move as they are satisfied less easily. This also shows in the increasing number of steps and non-terminating runs, as more moves are required to satisfy the agents. One last phenomenon that is difficult to explain is that segregation increases before it decreases.

5.4.2 Data Preparation

Based on the previous aspects, it was possible to define the input parameters that were varied for the data collection of the ABS. The final parameter ranges can be found in Table 4. These ranges were refined according to the parameter search. It came to be that size did not influence the segregation but did increase run-time significantly. Using 4 groups increased run-time as well, especially in the heterogeneous scenario, which is why only 2- or 3-group scenarios were included. The outputs of the ABS, as shown in Figure 8 are designed to capture the final state of the simulation.

Parameter	Value Range	Interval
Size	20x20	-
Number of Groups	2, 3	-
Density	0.2-0.99	0.1, 0.9 in last step
Heterogeneity	0.2	-
Agent Rule 1	0.2 - 0.9	0.1
Agent Rule 2	0.2 - 0.9	0.1
Agent Rule 3	0.2 - 0.9 or 0 if Number of Groups is 2	0.1
Maximum Number of Steps	100	-

Table 4. The final parameter ranges that were chosen for the simulation runs.

5.4.3 Modeling

The model implementation, experiment setup, and data collection are described here. A flowchart that depicts how the agent-based model is implemented in Agentpy and used for experiments can be found in Figure 9. Three loops can be found in the flowchart. The loop on the lowest level is the loop for simulation steps, where the agent behavior occurs. The higher level loop is based on iterations and records the information & resets the previous experiment. The highest level loop not only resets the experiment but also updates the parameters. Once all parameter settings are experimented on, the experiment is ended and the data is exported. The setup of the model and move_location event rely on a random number generator. The seed values for this generator are kept the same across all samples for the sake of comparability, although each iteration within such a sample does have a unique seed.

As seen in Figure 9, a set of activities is recorded in both the model update and step functions. The first activity is recorded when an agent changes to a happy state and the second activity is recorded when an agent moves to an unhappy state. The third activity occurs in the model step for unhappy agents when it moves to a random empty cell in the grid. The naming of cases in the event log adheres to the schema *activity_x_y* for happiness status changes. Here, x refers to the number of first-degree neighbors and y to those that belong to the same group. This was done for the sake of increasing complexity and variation between different experiments: event logs are likely identical or very similar if only the activity is recorded. It was found that this naming schema for a moving agent activity generated process models that were too large for evaluation, thus it is only known as *move_location*. Events do not have an associated time duration: only the step of each event is recorded, where each step is considered 1 second long (though it could be anything here, as long as it is consistent for all events) such that the logs are chronological.

The above covered the model implementation, setup, and most of the data collection. Lastly, we will briefly look into what "record outcome" refers to in Figure 9. Not only does it store the event log in a *.xes* file, but it also aggregates the model outcomes and adds them to the Agentpy experiment outcomes as a list of reporters. These reporters contain both the associated inputs (i.e., all input parameters, the seed, and the iteration) as well as numerical outputs

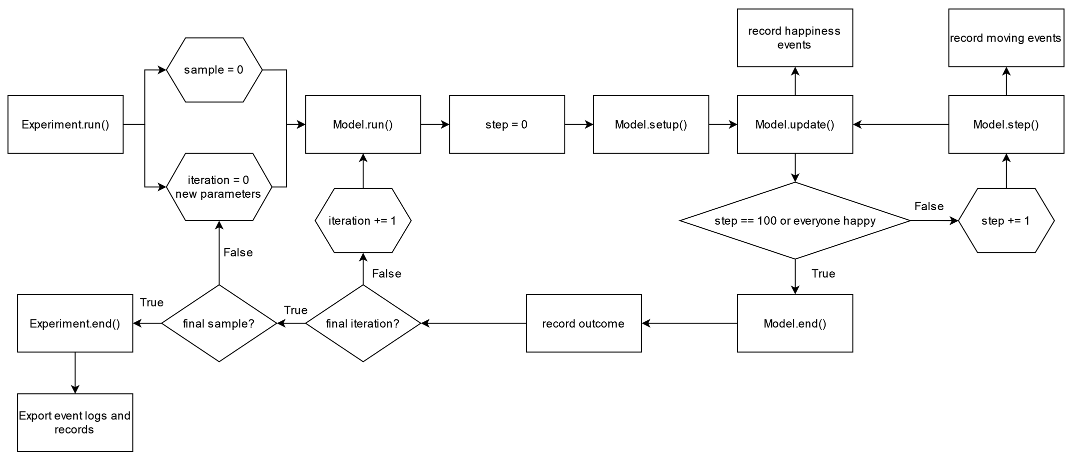

Figure 9. Experiment implementation.

5.4.4 Evaluation

The evaluation consisted of a brief analysis, to expand on the aspects of the business & data understanding phase, and a recounting of all the data points as described in Figure 8. All features were present in the final data set and 25.920 runs were performed, of which 9540 were unique. Duplicates are explained and removed in data preparation phases later on. Based on this, it can be evaluated if the objectives are achieved. The first objective was to collect event logs and information about ABM performance. This objective was completed, as event logs were successfully captured. The second objective was to gain insight into how segregation between agents is affected by topology and the agent rule, which was also achieved through the parameter search.

5.5 Process Mining Cycle

5.5.1 Business and Data Understanding

First, the PM objectives are to be determined after which the PM techniques are decided on. The objectives of this cycle are phrased as follows:

- to determine which techniques are appropriate for extracting more data from Schelling's model;
- to extract process models from previously generated event logs in the shape of a Petri net; and
- to extract information from those process models.

The heuristic miner was chosen as the discovery algorithm. The alternatives for procedural process model discovery available in this library are the alpha, alpha plus, integer linear programming, and inductive miners. The alpha miner was ruled out due to being unable to deal with loops. The alpha plus miner can deal with short-loops and self-loops, but initial experiments have shown that the resulting Petri nets were not usable. Other mining techniques such as the inductive miners, generational miners, and fuzzy miners are likely to be viable, but the heuristics miner was chosen due to its age and surrounding body of literature [85]. In addition, the heuristic miner is relatively good at dealing with noise and therefore likely to be used in practical situations as well. The heuristic miner was used with no additional input parameters or preferences for performance metrics. It was used to mine each event log in its entirety. In addition to this, it was decided to use token-based replay (TBR) to calculate any conformance metrics. Another possibility would be the use of alignments for more accurate metrics. However, this is too slow for a large number of event logs, even if an alignment recomposition technique is used [40].

The Pm4py documentation and papers that cover the techniques' implementation were used for the examination of the chosen PM techniques [56, 85]. The heuristic miner relies on, as the name suggests, heuristics of events for the creation of the process model from the event log. These heuristics can include event frequency and order of the events, among others. The benefit of this approach is that the discovery algorithm discovers the most probable sequences of events, which in turn allows for gaining previously unknown process knowledge and getting a grasp of the dynamics of the ABS model.

The TBR for conformance metrics is based on Rozinat and van der Aalst [63]. It compares the observed behavior recorded in event logs with the expected behavior described by the Petri net model. The algorithm replays the sequence of events from the event log on the Petri net by moving tokens along the model's places and transitions, verifying if the observed behavior aligns with the model. Misalignments and alignments between the two are used to calculate the metrics themselves.

5.5.2 Data Preparation

The first part of this phase is to define the PM outputs in line with the objectives and ABS cycle outputs. The first output would be a process model based on the heuristics miner. Information about the shape of the event log can be collected based on this, such as the number of arcs, places, and transitions in the Petri net. Using token-based replay, it is possible to calculate the fitness and precision of the Petri net on the original event log as well. Fitness was included, but precision was not due to its high computational complexity. Problems like this can probably be solved by GPU support in PM tool kits [9]. Even though fitness and precision are more akin to verification (and not validation) metrics, it is still expected they contain information. This is because one can argue, based on the ABS cycle, that an invalid non-terminating ABS model is more likely to have more logs covering every possible trace. This increases the likelihood of the traces existing in the heuristic miner-based model, thus making it likely higher conformance metrics could belong to "less valid" models. The final outputs can be found in Figure 8.

More types of output for the PM part are possible, but not considered. Some, such as cycle times and case duration, are not possible with this implementation of the Schelling model. Others, such as the previously mentioned precision, or alignment-based statistics, are computationally expensive. Additionally, it is possible to add additional attributes to collect event logs that can extract even more data, such as the previous state or information about how an agent's decision came to be. These statistics and many more are available in the "stats" package of the Pm4py library [56].

A few steps were performed to prepare the PM input. The first was to create a naming convention that describes each unique event log, which is also associated with both the ABS as well as the PM outputs in order to easily merge the data later. Secondly, one can use the mining algorithm and store the Petri nets before extracting information from the process models. This is beneficial if one expects to do multiple iterations on complex event logs so that each event log does not need to be mined more than once. This was not done, as complexity was kept at bay in this case study, and process mining was not expected to significantly impact calculation times.

Lastly, duplicates in the simulation results had to be removed. These duplicates occur because of the agent rule combinations: two heterogeneous 2-agent experiments with thresholds at (0.2, 0.3) and (0.3, 0.2) are essentially the same. Each first occurrence of such a combination was kept (e.g., (0.2, 0.3) in this case). Samples after this are removed. This was done in order to keep the data proportionate and reduce calculation times. After defining the objectives and the outputs, and preparing the inputs, it is now time to move on to modeling.

5.5.3 Modeling

As previously stated, everything was implemented in Pm4py. The steps in the PM cycle implementation are as follows:

1. Use the heuristic miner to get the initial marker, final marker, and Petri net for any event log.
2. Extract the number of places, transitions, and arcs from the Petri net.
3. Use TBR to determine the fitness.
4. Add information to a data frame and repeat for the following event logs.

Implementing this was part of step 14 in the approach. The first action relates to step 15. The remaining actions are part of the data extraction from the process model and thus belong to step 16. The implementation in Pm4py and pandas as opposed to tools such as Disco or ProM allows for easily adding or removing data collected in the data frame and merging it with the ABS data in the DM cycle.

5.5.4 Evaluation

Descriptive statistics were used to determine whether or not the collected data adhered to the defined PM outputs, which was the case. After this, the objectives were re-evaluated. It can be said that objective 1 is completed in the business and data understanding steps and objectives 2 and 3 in the modeling phase because Petri nets from the event logs are collected and information is extracted from this Petri net. The quality of the collected process models was high: some basic information regarding trace fitness can be found in Table 5. An example of such a Petri Net can be found in Appendix B, Figure 15.

Statistic	Fitness
Minimum	32.83%
Maximum	100%
Median	99.89%
Standard Deviation	$\pm$2.51%

Table 5. Petri net quality according to the calculated conformance metrics over all experiments.

5.6 Data Mining Cycle

5.6.1 Business and Data Understanding

The initial step in this phase is to motivate the DM objectives. These are

- to create an easily interpretable DM model;
- to find out what an optimal combination of features is with respect to performance and interpretability;
- to find how the decision trees react to different combinations of features and a lower amount of data

The last objective has the goal of demonstrating how the approach generalizes.

Now, the classification techniques and metrics are determined. The first objective is easiest to complete with a decision tree model, especially when it is kept shallow. The second objective can be completed with a wide range of feature selection techniques. In this case, the data is to be analyzed briefly by performing a principal component analysis (PCA) after which model-based feature selection is to be used. The third objective is done by using different combinations of input features and running an experiment with a low amount of data.

5.6.2 Data Preparation

Step 21 describes the formatting and cleaning of the data. First, the data sets from the ABS and PM cycles were joined with their unique file names as keys. Once again, a check for duplicate data was done. Outliers were not checked for extensively and kept, as descriptive statistics did not show any and the relatively large amount of data points would make up for this.

After this, it was necessary to determine how the target variable is to be calculated. Note that if one decides to normalize the data, it is to be calculated before doing so. The target variable is calculated based on a movement-oriented metric:

$$M = \frac{m_l}{m_f}$$

Here M refers to the metric, m_f to the number of moves in the first step of the experiment, and m_l to the number of moves in the last experiment. The metric is designed on the assumption that when Schelling's model functions properly, it converges. For this metric, a value approaching 0 suggests the number of moves has declined and the model converges. A value closer to 1 suggests that the number of moves, and by extension the number of unhappy agents, has remained constant. As validity is a binary classification problem, a threshold for this metric is used to classify a model as valid or invalid. This threshold was based on a real-world number in order to adhere to the main goal. The number of people that move to a different home on a yearly basis in the Netherlands was chosen, which is 9.3% according to the CBS [17] (i.e., a Dutch government organization that is responsible for data collection). Thus, a "valid model" would be when a model terminates or is close to terminating, representing a 1. Vice versa is classified as 0. This real-world number was chosen because the segregation model represents people moving, as described by Schelling [66]. In this case, 100 steps could for example represent a year. An important note is that the model does not represent the real world and that this threshold is chosen for demonstrative purposes.

Following the target cleaning and target variable steps, it was possible to split the data sets, which was done in the following fashion. The experiments were randomly distributed in such a way that all iterations per sample were grouped together, but samples were not. This is necessary for both k-fold cross-validation as well as splitting of the train and test set. The train set contained 80% of the data, and the test set 20%. In the smaller data set used in the experiments, the train set contained 10% of the initial training data set (so 8%) and the test set remained at 20%.

Finally, it was possible to perform feature selection on the training set. This was done through principal component analysis and the "select from model" algorithm in the Scikit-learn library, both of which were covered in the first step of this cycle.

The PCA can be found in Figure 10. One can see that the classes are only partially separated in the second dimension, which is sub-optimal but not bad. There are two clear separations though, which are caused by the number of groups in the simulation. This is probably because the third agent rule is kept at 0 when there is no third agent group. The PCA showed that the first principal component is mostly explained by the Petri Net: the number of places, arcs, and transitions. The second principal component is explained mostly by the number of groups, the number of agents, and the number of steps. It is interesting to see that the PCA yielded different results than the feature selection. However, due to the sub-optimal separation, it was decided not to use the features from this technique in the experiments.

Three features were selected by the "Select From Model" method in the Scikit-learn library. The method is a meta-transformer, which assigns importance to each feature. The meta-transformer works with any classification method, but a set of 50 decision trees was used since the main classification method also consists of a decision tree. Running it on all possible inputs resulted in the final segregation, number of logs, and number of steps being selected.

5.6.3 Modeling

The modeling phase consists of training the models and subsequently tuning the model parameters. These two steps are iterated until the result is satisfying. In this scenario, the parameters were left at their defaults for most experiments. The only model was a decision tree, but the amount of data and input were modified to get an idea of how well the approach generalizes. The trees were implemented using Scikit-learn, which makes use of an optimized version of the CART algorithm. Experiments 5 and 11 also had a limitation to the depth of the tree: a smaller tree is often more easily interpreted by humans and is valuable with respect to the objective. The experiments were executed with 10-fold cross-validation. The results of the modeling steps can be seen in Table 6. An in-depth analysis of these results is saved for the results section. However, the models perform extremely well and there is hardly any significant difference between different feature combinations. Even the small data set generalizes well, which shows that the approach is likely to work in cases with little data available too. One noticeable drop in performance exists for the PM inputs, which suggests these might not contribute much to the decision-making of the model.

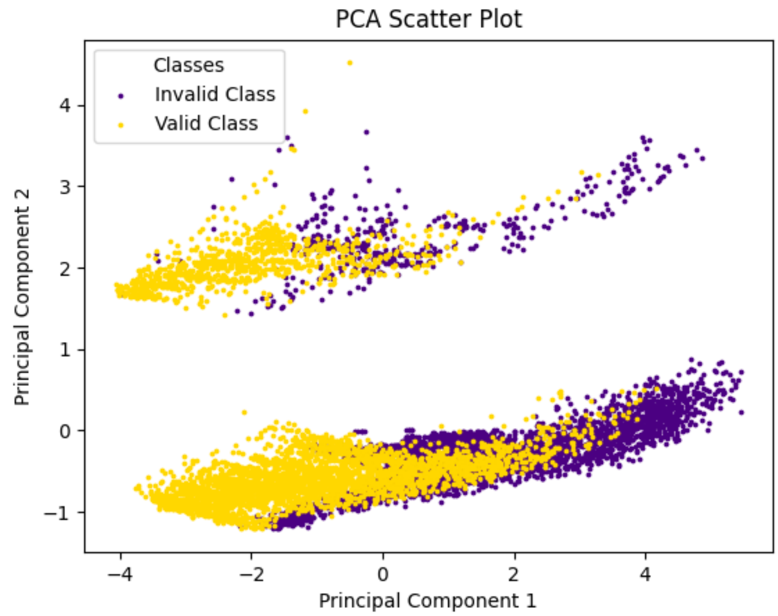

Figure 10. Principal Component Analysis of the data for two components.

The small tree with a depth of 3 is the preferred output of this phase, as it is likely to be interpretable and only uses a small set of features. Additionally, the performance is not significantly lower.

Experiment	Model	Data Set	Mean accuracy ($\pm\sigma$)	Mean F1-score ($\pm\sigma$)
0	abs_inputs	normal train	0.93 ($\pm$0.01)	0.93 ($\pm$0.01)
1	pm_outputs	normal train	0.82 ($\pm$0.02)	0.83 ($\pm$0.02)
2	abm_pm_outputs	normal train	0.98 ($\pm$0.00)	0.98 ($\pm$0.00)
3	all	normal train	0.98 ($\pm$0.01)	0.98 ($\pm$0.01)
4	feature_selection	normal train	0.96 ($\pm$0.00)	0.96 ($\pm$0.01)
5	feature_selection_small	normal train	0.97 ($\pm$0.01)	0.97 ($\pm$0.01)
6	abs_inputs	small train	0.90 ($\pm$0.06)	0.90 ($\pm$0.07)
7	pm_outputs	small train	0.78 ($\pm$0.04)	0.78 ($\pm$0.07)
8	abm_pm_outputs	small train	0.96 ($\pm$0.02)	0.96 ($\pm$0.02)
9	all	small train	0.96 ($\pm$0.02)	0.96 ($\pm$0.03)
10	feature_selection	small train	0.95 ($\pm$0.04)	0.96 ($\pm$0.03)
11	feature_selection_small	small train	0.97 ($\pm$0.03)	0.97 ($\pm$0.03)

Table 6. Results from the modeling phase of the case study.

5.6.4 Evaluation

The first step here is to calculate the metrics on the test set and evaluate the performance. These numbers can be found in Table 7. While the smaller data set is not relevant for this phase, it is still interesting to see that it performs worse for both the ABS inputs and the PM outputs, but that the remaining experiments perform only slightly worse or comparable to the larger data sets. The feature selection for these models also resulted in 5 as supposed to 3 features previously. This suggests more input features are required for a well-performing result, but that the approach still performs well. The optimal model for deployment is experiment 5: the tree is shallow, it performs well on the test set and the number of features is low.

Finally, the objectives are evaluated. The first objective regarding interpretability is achieved. As can be seen in 11, it is relatively simple to read how an ABS is classified by the model. The tree will be covered extensively in the discussion, which is found in Section 6. The optimal set of features according to both experiments seems to be all features combined, but using just the number of logs, number of events and the segregation provides almost the same results in the experiments. Thus, the second objective is also achieved. The last objective was also achieved, as it can be seen that certain feature sets and a lower amount of data slightly increase the variance, accuracy, and f1-score even though performance is still quite good.

Experiment	Model	Data Set	Accuracy	F1-score
0	abs_inputs	normal train and test	0.90	0.91
1	pm_outputs	normal train and test	0.85	0.86
2	abm_pm_outputs	normal train and test	0.97	0.98
3	all	normal train and test	0.97	0.98
4	feature_selection	normal train and test	0.96	0.96
5	feature_selection_small	normal train and test	0.96	0.96
6	abs_inputs	small train and test	0.85	0.86
7	pm_outputs	small train and test	0.75	0.76
8	abm_pm_outputs	small train and test	0.94	0.95
9	all	small train and test	0.94	0.94
10	feature_selection	small train and test	0.96	0.97
11	feature_selection_small	small train and test	0.95	0.95

Table 7. Results of the final models on the test set.

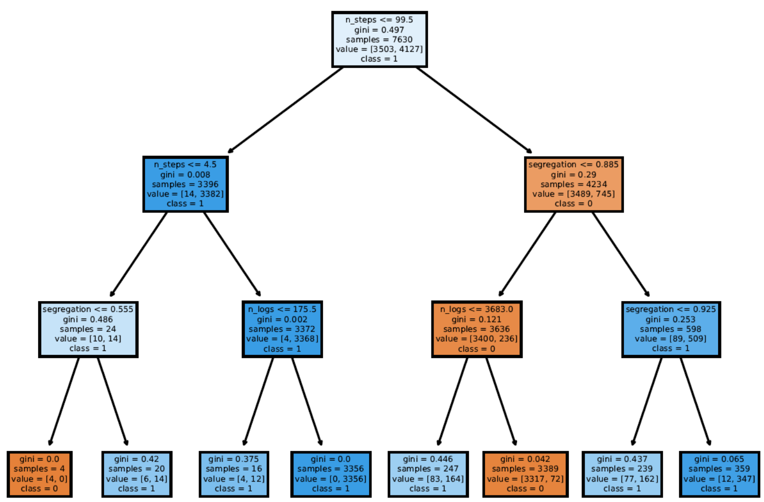

Figure 11. Decision tree from experiment 5.

5.7 Deployment

The deployment was not part of this case study, as there is no real-world use case. Therefore, this section will briefly describe how it could hypothetically be implemented.

First, the user will have to fill in the new parameter settings for the ABS to provide the new ABS model. This triggers the start of the validation process, which is automated. A simulation experiment that outputs the same information as in Figure 8 is performed. The resulting event logs are immediately mined using Pm4py. Both the ABS and PM results are joined together in a single data frame. After this, the data corresponding to the necessary input features derived previously from the feature selection (number of logs, number of events, the final segregation) are inserted into the existing decision tree model. Finally, the output of the program to the user will be whether or not the model is valid and the probability score of the output.

This approach can then be applied in combination with other validation approaches in order to achieve a higher certainty of the correctness of the model output.

5.8 Concluding Remarks

This subsection aims to briefly cover the fourth research question. It is as follows: "How can this implementation for the validation approach be operationalized?"

This case study describes that the approach can be operationalized by following the steps resulting from the previous research question, as presented in Subsection 4.3.

6 DISCUSSION

This discussion section serves the purpose of evaluation as defined in the DSRM by Peffers et al [54]. First, a brief discussion on the objectives is done. It relies on logical, case-based inference as well as the metrics resulting from the case study. A brief summary of the objectives for the artifact is as follows:

- Interdisciplinary: the approach has to combine the fields of ABS, PM, and DM successfully.
- Adaptability: the components in the approach have to be replaceable (e.g., replace one PM algorithm with another).
- Efficacy: the approach has to be effective in dealing with the validation problem.
- Interpretability: the approach has to be interpretable.

6.1 Objectives

6.1.1 Interdisciplinary

The first objective to be judged is whether the approach was successfully interdisciplinary or not. This will be evaluated based on the feature selection steps, which can represent how valuable and collaborative the fields of ABM, PM, and DM are in the proposed approach and the case study.

To start with, we can judge part of their presence through the feature selection step in the data mining cycle. According to the principal component analysis in the data mining cycle, it seems that most of the variance on the move metric was explained by process mining results. However, the PCA did not show a particularly good separation of the data. The regular feature selection results mainly featured ABS cycle results, including the number of events as this is calculated during the modeling phase. This would mean process mining only contributed marginally to the goal of validating the agent-based simulation model. However, process mining does help with interpretability by generating a process model such as in Figure 15 in Appendix B.

Finally, Figure 12 shows how each cycle and discipline affected each other, as well as the goal. The solid arrows indicate a strong dependency, and the dotted arrow indicates a medium dependency. We can conclude that this objective was mostly achieved, as 2 out of 3 disciplines contributed greatly and collaborated, whereas the last discipline partially contributed.

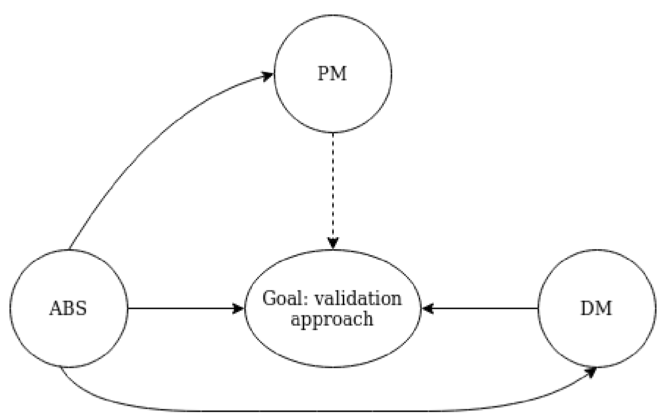

Figure 12. Figure showing where the disciplines collaborated towards the goal of validation of an agent-based simulation model.

6.1.2 Adaptability

The second objective refers to adaptability. As can be seen in Figure 7 for the approach and Figure 8 for the exchanged information in the case study, each cycle is disjoint from the previous techniques. The only constraints are the types of information shared between each cycle, i.e., process mining requires an event log and data mining requires the previously generated features as well as a target variable. This means that any technique used can be replaced with another as long as the output stays the same: one can easily replace Agentpy in the agent-based simulation cycle with Netlogo or Repast for example. Because the cycles and techniques used are replaceable, we can conclude that adaptability has been achieved.

6.1.3 Efficacy
The third objective corresponds to whether the solution is effective in dealing with the problem. We will first look at the empirical data from the case study and then reason through logic.

The process mining cycle yielded high-quality Petri nets, as can be seen from Table 5. This would generally be a good sign and it does suggest that process mining is effective for the Schelling model. Meanwhile, the median is extremely high and the standard deviation is low. This means that there is a heavy skew in the data and that the variance is low; which also suggests that the contribution of the process mining cycle is small.

The data mining cycle also yielded high accuracy and F1-scores for most feature combinations and both levels of data availability, as can be seen in Table 6 and Table 7. This suggests the data mining cycle, and by extension on the test results also the approach, was effective in determining whether an agent-based simulation model belongs to the valid or invalid class.

One weakness of the efficacy is that the target variable, which is supposed to resemble validness, is relatively simple in the case study. Since it is based on model outputs (which in turn depend on model inputs), there is likely to be a form of dependency that makes classification easier. It is unknown how the efficacy applies to a more complicated ABM where valid models are determined through for example a survey or expert opinion.

Therefore, it can be said that the efficacy objective is completed for the case study and similar use cases. The generalization of efficacy on more complex models and validation definitions is not as clear and requires future research.

6.1.4 Interpretability
The final objective is interpretability for humans. There are multiple components of the approach that facilitate interpretability that will be briefly mentioned.

The first is the process model. It helps one understand certain mechanisms of an agent-based simulation model. For example (from Figure 15 in Appendix B): one can deduce that there is a set of happiness events that seemingly always lead to termination, but that there exist other sets of happiness events that can iterate back to unhappiness events.

A second tool for interpretability is the decision tree from the case study as shown in Figure 11. It allows the implementer of the approach to understand how an agent-based model is classified, which can lead to a better understanding of a valid or invalid model, parameter, or validation metric. It also allows one to obtain a probability of either validity or invalidity using Scikit Learn's probabilistic prediction, which also forms the foundation of the usual prediction function in the library [53, 68].

A third tool is the principal component analysis. It shows which variables have the most variance with respect to each principal component, and it allows one to visualize data separation. In the case of Figure 10, it can be seen that there is a large difference between the two clusters. Additional analysis has shown that this is caused by the number of agent groups in the model.

However, interpretability depends highly on the tools chosen in the approach. If one chooses to add a visualization for simulations, it would be possible to increase it. If one opts to use an artificial neural network for classification, one would decrease it.

Based on the above, it can be said that the interpretability objective of the approach is achieved. While there is still room for improvement, the approach already provides a multitude of interpretable tools.

6.2 Limitations
This subsection discusses the limitations of the study with respect to the approach and the case study.

6.2.1 Single Variable
The first limitation is with respect to the definition of validity in this study and the use of a single metric to define it in the case study. In order to overcome this limitation, one can consider marking several agents as valid or invalid by hand, perhaps verified by expert opinion. The approach has proved to generalize quite well with smaller data sets, so it is expected that this would address the problem.

6.2.2 Heuristic Miner

The second limitation involves the use of the heuristic miner and conformance metrics as features in the case study. Although it is a well-researched and documented algorithm that often yields good results, it might not be suitable for this scenario. The heuristic miner is designed to achieve high heuristic results independent of the log and model quality. This means that there is little variance between the data points collected from this algorithm, which in turn limits the information gained. In future research, one can consider both the inductive and fuzzy miners. However, the latter is not available in the Pm4py library.

6.2.3 Context

A third limitation is the context of this research: the topology, heterogeneity, and agent rules. This is only a limited set of input parameters, and it is tested on a simple ABS model. Although the approach is designed to generalize, it cannot be said with certainty that the validation results hold up for complex ABS models that are changed often and drastically. In order to address this limitation, one can consider including more concepts mentioned in Subsection 2.1.4 for another case study on a more elaborate ABS model.

6.2.4 Design Cycle

Only one iteration of the development and evaluation of the artifact was performed. Even though it is optional, the approach can probably be refined with respect to the lower value of process mining and additional steps that guide interpretability in more detail. Once this is done, another evaluation could yield a more comprehensive overview of the efficacy of the approach as well.

6.2.5 Validity

The final limitation that is discussed in this section is how validity is defined. This thesis follows a definition that compares a model with respect to real-world elements and tailors to the concept of objective validity. However, it partially accounts for the philosophical aspect of validation by including interpretability as an objective and allowing a degree of freedom for tailoring the definition of validation for each model to which the approach is applied. The downsides are that this narrow definition tailored to the Schelling model might limit the generalizability of the approach and that the narrow definition with respect to accuracy does not account for usefulness. Moreover, the interpretability of the approach only provides a limited understanding of how model behavior comes to be, implying that there might be a need for expansion on this topic in order to properly validate white-box models.

6.3 Concluding Remarks

This subsection aims to answer the fifth research question. The fifth question is as follows: "Does the proposed validation approach for agent-based simulation models solve the research problems?"

This question can be answered by considering the discussed research objectives, which were derived from the problem statement in Subsection 1.2. All objectives were largely completed, even though this section has also shown that there is still some room for improvement. Considering the extent to which extent the objectives were achieved, it can be said that the validation approach for agent-based simulation models was indeed able to solve the research problems.

7 CONCLUSION AND FUTURE RESEARCH

This section aims to first briefly recap the answers and relate them to the research questions. After this, the main question is answered. Finally, directions for future research are provided.

7.1 Conclusion

7.1.1 Research Questions

This subsection first lists the relevant research questions with a short answer. After this, a detailed description of the answers is given based on the conclusive remarks of their corresponding sections. The research questions were as follows:

Question 1: What are the key factors and definitions when assessing the validity of an agent-based simulation model in the context of the main research question?

Answer: The answer is based on structured literature research and the contents of Section 2. The first key factor consists of agent-based simulation, which is the process of designing an agent-based model for a real system and conducting experiments with this model [69]. After this, included concepts and excluded concepts were discussed in Subsection 2.1.3 and the subsequent subsection. The second key factor is process mining, a data-driven technique that aims to extract valuable insights from event logs [81]. The third factor which is key to this work, is data mining with a focus on classification. Data mining is another data-driven technique for knowledge discovery, generally aimed at extracting valuable information from raw data [30]. Lastly, the concept of validity is important. Validity in this work is assumed to be the degree to which a model is an accurate representation of a real-world system that it is intended to simulate [61, 65], though alternative definitions exist.

Question 2: What are suitable methodologies for the development of a validation approach for agent-based simulation models?

Answer: Based on the considerations in Section 3, it was decided that a suitable methodology for the development is the DSRM. Additionally, it was found that CRISP-DM is a suitable foundation for the validation approach.

Question 3: How can this methodology be implemented for the validation approach?

Answer: A detailed description of this question can be found in Section 4. The section found that a three-cycle approach based on CRISP-DM is a solid foundation for implementing such a validation approach, and that is preferable over a single-cycle interpretation. Four objectives were designed for the implementation of the validation approach: interdisciplinary, adaptability, efficacy, and interpretability.

Question 4: How can this implementation for the validation approach be operationalized?

Answer: The case study in Section 5 describes that the approach can be operationalized by following the steps resulting from the previous research question, as presented in Subsection 4.3. The section successfully demonstrates the approach, as a case study, on the segregation model by Schelling [66].

Question 5: Does the proposed validation approach for agent-based simulation models solve the research problems?

Answer: The final question can be answered by considering the discussed research objectives, which were derived from the problem statement in Subsection 1.2. As shown in Section 6, all objectives were largely completed. However, the discussion has also shown that there is still some room for improvement with respect to the use of a single metric, the heuristic miner, the research context, the design cycle, and the concept of validity. Considering the extent to which the objectives were achieved, it can be said that the validation approach for agent-based simulation models was indeed able to solve the research problems.

7.1.2 Main Research Question

This subsection will answer the main question based on the preceding research questions. The main question is presented as follows:

"How can an agent-based simulation model's validity be assessed using process mining and data mining techniques in the context of the model's topology, heterogeneity, and agent behavior rules?"

The validity of an agent-based simulation model can be assessed using process mining and data mining in the context of a model's topology, heterogeneity, and agent behavior rules. This was achieved by first completing a structured literature review and expanding on important concepts as per the first research question. After this, the answers to research questions 2 and 3 demonstrate how an approach that combines process mining and data mining can be developed. Question 4 was answered with a successful case study and question 5 evaluated it, which found that the objectives of the approach were achieved within the research context.

7.2 Contribution to Science

This thesis contributes to the field of agent-based model validation. It shows how agent-based simulation models, process mining, and data mining can be combined for an objective validation approach based on a three-cycle CRISP-DM interpretation. This approach supports improved maintainability, interpretability, and validity of agent-based simulation models, solving a number of research challenges identified in Subsection 2.6. Additionally, it paves the way for future research for the combination of agent-based simulation models, process mining, and data mining.

7.3 Future Research

The first direction of future research is to generalize the approach for agent-based models, and not just agent-based simulation models. This can be done in part by addressing the limitations in Subsection 6.2 and performing more complex, real-world case studies.

A second possibility is the integration with other validation strategies, such as face validation with the help of the interpretable aspects of the approach. Validation consists of many aspects, and combining these with the goal of eventually automating a large part would be beneficial.

Finally, future research into further integrating ABS, PM, and DM is suggested. Two examples of this are the use of existing event logs to automatically generate a conforming agent-based simulation model or to evaluate agent-based model behavior using data mining in an interpretable manner.

REFERENCES

[1] M. Abdou, L. Hamill, and N. Gilbert. Designing and building an agent-based model. *Agent-Based Models of Geographical Systems*, pages 141–165, 1 2012. doi: 10.1007/978-90-481-8927-4{_}8/FIGURES/4{_}8. URL `https://link.springer.com/chapter/10.1007/978-90-481-8927-4_8`.

[2] A. K. Alves De Medeiros, B. F. Dongen van, W. M. P. Aalst van der, and A. Weijters. *Process mining : extending the alpha-algorithm to mine short loops*. BETA publicatie : working papers. Technische Universiteit Eindhoven, 2004. ISBN 90-386-1978-2.

[3] F. P. Appio, M. G. C. A. Cimino, A. Lazzeri, A. Martini, and G. Vaglini. Fostering distributed business logic in Open Collaborative Networks: an integrated approach based on semantic and swarm coordination. *Information Systems Frontiers*, 20(3):589–616, 6 2018. ISSN 1387-3326. doi: 10.1007/s10796-016-9691-5.

[4] O. Baqueiro, Y. J. Wang, P. McBurney, and F. Coenen. Integrating data mining and agent based modeling and simulation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5633 LNAI:220–231, 2009. ISSN 03029743. doi: 10.1007/978-3-642-03067-3{_}18/COVER. URL `https://link-springer-com.ezproxy2.utwente.nl/chapter/10.1007/978-3-642-03067-3_18`.

[5] M. Barbati, G. Bruno, and A. Genovese. Applications of agent-based models for optimization problems: A literature review. *Expert Systems with Applications*, 39(5):6020–6028, 4 2012. ISSN 0957-4174. doi: 10.1016/J.ESWA.2011.12.015.

[6] Y. Barlas. Formal aspects of model validity and validation in system dynamics. *System Dynamics Review*, 12(3):183–210, 1996. doi: 10.1002/(SICI)1099-1727(199623)12:3. URL `https://onlinelibrary.wiley.com/doi/10.1002/`.

[7] R. H. Bemthuis and S. Lazarova-Molnar. Discovering Agent Models using Process Mining: Initial Approach and a Case Study. In *Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking*, pages 163–172, 2020. doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00028.

[8] A. Berti, S. V. Zelst, and W. M. v. d. Aalst. Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science. *ArXiv*, 2019.

[9] A. Berti, M. P. Nghia, and W. M. P. van der Aalst. PM4Py-GPU: a High-Performance General-Purpose Library for Process Mining. 4 2022. URL `http://arxiv.org/abs/2204.04898`.

[10] F. Bianchi and F. Squazzoni. Agent-based models in sociology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(4):284–306, 7 2015. ISSN 1939-0068. doi: 10.1002/WICS.1356. URL `https://onlinelibrary.wiley.com/doi/full/10.1002/wics.1356https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.1356https://wires.onlinelibrary.wiley.com/doi/10.1002/wics.1356`.

[11] A. Bogarín, R. Cerezo, and C. Romero. A survey on educational process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(1):e1230, 1 2018. ISSN 1942-4795. doi: 10.1002/WIDM.1230. URL `https://onlinelibrary-wiley-com.ezproxy2.utwente.nl/doi/full/10.1002/widm.1230https://onlinelibrary-wiley-com.ezproxy2.utwente.nl/doi/abs/10.1002/widm.1230https://wires-onlinelibrary-wiley-com.ezproxy2.utwente.nl/doi/10.1002/widm.1230`.

[12] J. D. Bohlmann, R. J. Calantone, and M. Zhao. The Effects of Market Network Heterogeneity on Innovation Diffusion: An Agent-Based Modeling Approach. *Journal of Product Innovation Management*, 27(5):741–760, 9 2010. ISSN 1540-5885. doi: 10.1111/J.1540-5885.2010.00748.X. URL `https://onlinelibrary-wiley-com.ezproxy2.utwente.nl/doi/full/10.1111/j.1540-5885.2010.00748.x`.

[13] E. Brzychczy. An Overview of Data Mining and Process Mining Applications in Underground Mining. *Inżynieria Mineralna*, R. 21, nr 1(1):301–314, 2019. ISSN 1640-4920. doi: 10.29227/IM-2019-01-52. URL `http://doi.org/10.29227/IM-2019-01-52`.

[14] Cambridge University Press. HETEROGENEITY | English Meaning. URL `https://dictionary.cambridge.org/dictionary/english/heterogeneity`.

[15] K. M. Carley, D. B. Fridsma, E. Casman, A. Yahja, N. Altman, L. C. Chen, B. Kaminsky, and D. Nave. BioWar: Scalable agent-based model of bioattacks. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 36(2):252–265, 2 2006. ISSN 10834427. doi:

10.1109/TSMCA.2005.851291.

[16] Celonis. Process Mining and Execution Management Software. URL `https://www.celonis.com/`.

[17] Centraal Bureau Statistiek. Verhuizen. URL `https://www.cbs.nl/nl-nl/visualisaties/dashboard-bevolking/levensloop/verhuizen`.

[18] N. Collier and J. Ozik. Distributed Agent-Based Simulation with Repast4Py. *Proceedings - Winter Simulation Conference*, 2022-December:192–206, 2022. ISSN 08917736. doi: 10.1109/WSC57314.2022.10015389.

[19] M. Cremonini, A. Omicini, and F. Zambonelli. Multi-agent systems on the internet: Extending the scope of coordination towards security and topology. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1647:77–88, 1999. ISSN 16113349. doi: 10.1007/3-540-48437-X{_}7/COVER. URL `https://link-springer-com.ezproxy2.utwente.nl/chapter/10.1007/3-540-48437-X_7`.

[20] B. F. Dongen van, J. Luin van, and H. M. W. Verbeek. Process mining in a multi-agent auctioning system. In D. Moldt, editor, *Proceedings of the Fourth International Workshop on Modelling of Objects, Components and Agents, MOCA'06 ; [26th June, 2006, Turku, Finland]*, Bericht Universität Hamburg, Department Informatik, pages 145–160. University of Hamburg, 2006.

[21] B. Edmonds, C. Little, L. Lessard-Phillips, and E. Fieldhouse. Analysing a complex agent-based model using data-mining techniques. *Social Simulation Conference*, 2014. URL `https://ddd.uab.cat/record/128002`.

[22] A. Fleischmann, U. Kannengiesser, W. Schmidt, and C. Stary. Subject-oriented modeling and execution of multi-agent business processes. In *Proceedings - 2013 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2013*, volume 2, page 138 145, 2013. doi: 10.1109/WI-IAT.2013.102. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-84893310152&doi=10.1109%2fWI-IAT.2013.102&partnerID=40&md5=c17db2acf2465b0e127e8fcb7332a21f`.

[23] Fluxicon Disco. Process Mining and Automated Process Discovery Software for Professionals. URL `https://fluxicon.com/disco/`.

[24] J. Foramitti. AgentPy: A package for agent-based modeling in Python. *Journal of Open Source Software*, 6(62):3065, 6 2021. ISSN 2475-9066. doi: 10.21105/JOSS.03065. URL `https://joss.theoj.org/papers/10.21105/joss.03065`.

[25] R. Govers. Predicting Heart Rates Of Sport Activities Using Machine Learning. 2020.

[26] V. Grimm, S. F. Railsback, C. E. Vincenot, U. Berger, C. Gallagher, D. L. Deangelis, B. Edmonds, J. Ge, J. Giske, J. Groeneveld, A. S. Johnston, A. Milles, J. Nabe-Nielsen, J. G. Polhill, V. Radchuk, M. S. Rohwäder, R. A. Stillman, J. C. Thiele, and D. Ayllón. The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism. *2019:147:2*, 23(2), 3 2020. ISSN 1460-7425. doi: 10.18564/JASSS.4259. URL `https://www.jasss.org/23/2/7.html`.

[27] M. Halaška and R. Šperka. Is there a Need for Agent-based Modelling and Simulation in Business Process Management? *Organizacija*, 51(4):255–269, 12 2018. ISSN 1581-1832. doi: 10.2478/orga-2018-0019.

[28] M. Halaška and R. Šperka. Advantages of Application of Process Mining and Agent-Based Systems in Business Domain. In *Smart Innovation, Systems and Technologies*, volume 96, pages 177–186. Springer, 2019. doi: 10.1007/978-3-319-92031-3{_}17.

[29] M. Halaška and R. Šperka. Messaged Multi-agent System as a Tool for Strengthening Innovative Capabilities of Business Models. In *Agents and Multi-agent Systems: Technologies and Applications*, pages 355–365, 2019. doi: 10.1007/978-981-13-8679-4{_}29.

[30] D. Hand. Data Mining. *Encyclopedia of Environmetrics*, 10 2001. doi: 10.1002/9780470057339.VAD002. URL `https://onlinelibrary-wiley-com.ezproxy2.utwente.nl/doi/10.1002/9780470057339.vad002`.

[31] F. Hunka and S. van Kervel. A Generic DEMO Model for Co-creation and Co-production as a Basis for a Truthful and Appropriate REA Model Representation. In *Business Process Management: Blockchain and Central and Eastern Europe Forum*, volume 361, pages 203–218. Springer, 2019. doi: 10.1007/978-3-030-30429-4{_}14.

[32] IEEE. IEEE standard for extensible event stream (XES) for achieving interoperability in event logs

and event streams., 2016.

[33] S. Ito, D. Vymětal, R. Šperka, and M. Halaška. Process mining of a multi-agent business simulator. *Computational and Mathematical Organization Theory*, 24(4):500–531, 12 2018. ISSN 1381-298X. doi: 10.1007/s10588-018-9268-6.

[34] N. R. J E N N I N G S 3. On cooperation in multi-agent systems*. *The Knowledge Engineering Review*, 12(3):309–314, 1997. doi: 10.1017/S0269888997003111. URL https://doi.org/10.1017/S0269888997003111.

[35] N. R. Jennings, T. J. Norman, P. Faratin, P. O'Brien, and B. Odgers. Autonomous agents for business process management. *Applied Artificial Intelligence*, 14(2):145–189, 2000. doi: 10.1080/088395100117106. URL https://doi.org/10.1080/088395100117106.

[36] H. Jonkers, M. M. Lankhorst, H. A. Proper, D. Quartel, and M.-E. Iacob. ArchiMate(R) for Integrated Modelling Throughout the Architecture Development and Implementation Cycle. In *2011 IEEE Conference on Commerce and Enterprise Computing*, pages 294–301, 2011. ISBN 9780769545356. doi: 10.1109/CEC.2011.52. URL https://www.researchgate.net/publication/221542985.

[37] K. Kang, S. Belkhale, G. Kahn, P. Abbeel, and S. Levine. Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May: 6008–6014, 5 2019. ISSN 10504729. doi: 10.1109/ICRA.2019.8793735.

[38] S. Karamizadeh, S. M. Abdullah, A. A. Manaf, M. Zamani, and A. Hooman. An Overview of Principal Component Analysis. *Journal of Signal and Information Processing*, 4:173–175, 2013. doi: 10.4236/jsip.2013.43B031. URL http://www.scirp.org/journal/jsip.

[39] C. C. Kerr, R. M. Stuart, D. Mistry, R. G. Abeysuriya, K. Rosenfeld, G. R. Hart, R. C. Núñez, J. A. Cohen, P. Selvaraj, B. Hagedorn, L. George, M. Jastrzbski, A. S. Izzo, G. Fowler, A. Palmer, D. Delport, N. Scott, S. L. Kelly, C. S. Bennette, B. G. Wagner, S. T. Chang, A. P. Oron, E. A. Wenger, J. Panovska-Griffiths, M. Famulare, and D. J. Klein. Covasim: An agent-based model of COVID-19 dynamics and interventions. *PLOS Computational Biology*, 17(7):e1009149, 7 2021. ISSN 1553-7358. doi: 10.1371/JOURNAL.PCBI.1009149. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009149.

[40] W. L. J. Lee, H. M. Verbeek, J. Munoz-Gama, W. M. van der Aalst, and M. Sepúlveda. Recomposing conformance: Closing the circle on decomposed alignment-based conformance checking in process mining. *Information Sciences*, 466:55–91, 10 2018. ISSN 0020-0255. doi: 10.1016/J.INS.2018.07.026.

[41] S. J. Leemans, D. Fahland, and W. M. Van Der Aalst. Discovering block-structured process models from event logs - A constructive approach. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7927 LNCS:311–329, 2013. ISSN 03029743. doi: 10.1007/978-3-642-38697-8{_}17/COVER. URL https://link-springer-com.ezproxy2.utwente.nl/chapter/10.1007/978-3-642-38697-8_17.

[42] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. Feature selection: A data perspective. *ACM Computing Surveys*, 50(6), 12 2017. ISSN 15577341. doi: 10.1145/3136625. URL https://doi.org/10.1145/3136625.

[43] L. F. Lins, G. Melo, T. Oliveira, P. Alencar, and D. Cowan. PACAs: Process-Aware Conversational Agents. In *Business Process Management Workshops*, pages 312–318. Springer, 2022. doi: 10.1007/978-3-030-94343-1{_}24.

[44] S. Luke, C. Cioffi-Rivella, L. Panait, K. Sullivan, and G. Balan. MASON: A Multiagent Simulation Environment. *SIMULATION: Transactions of The Society for Modeling and Simulation International*, 81(7), 2005. doi: 10.1177/0037549705058073. URL https://www.researchgate.net/publication/220165043_MASON_A_Multiagent_Simulation_Environment.

[45] C. M. Macal and M. J. North. Agent-based modeling and simulation. *Proceedings - Winter Simulation Conference*, pages 86–98, 2009. ISSN 08917736. doi: 10.1109/WSC.2009.5429318.

[46] M. Molina, S. Carrasco, and J. Martin. CCIS 430 - Agent-Based Modeling and Simulation for the Design of the Future European Air Traffic Management System: The Experience of CASSIOPEIA. *CCIS*, 430:22–33, 2014.

[47] E. W. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun. The application of data mining techniques in

financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50(3):559–569, 2 2011. ISSN 0167-9236. doi: 10.1016/J.DSS.2010.08.006.

[48] J. Ni, L. Muhlstein, and J. McAuley. Modeling heart rate and activity data for personalized fitness recommendation. *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, pages 1343–1353, 5 2019. doi: 10.1145/3308558.3313643. URL `https://dl-acm-org.ezproxy2.utwente.nl/doi/10.1145/3308558.3313643`.

[49] R. Oberhauser and S. Stigler. Microflows: Leveraging Process Mining and an Automated Constraint Recommender for Microflow Modeling. In *Business Modeling and Software Design*, pages 25–48, 2018. doi: 10.1007/978-3-319-78428-1{_}2.

[50] C. C. Osman and A. M. Ghiran. When Industry 4.0 meets Process Mining. *Procedia Computer Science*, 159:2130–2136, 1 2019. ISSN 1877-0509. doi: 10.1016/J.PROCS.2019.09.386.

[51] C. Ou-Yang and Y.-C. Juan. Applying process mining approach to support the verification of a multi-agent system. *Journal of Systems Science and Systems Engineering*, 19(2):131–149, 6 2010. ISSN 1004-3756. doi: 10.1007/s11518-010-5132-z.

[52] M. H. Patel, M. A. Abbasi, M. Saeed, and S. J. Alam. A scheme to analyze agent-based social simulations using exploratory data mining techniques. *Complex Adaptive Systems Modeling*, 6(1):1–17, 12 2018. ISSN 21943206. doi: 10.1186/S40294-018-0052-8/FIGURES/8. URL `https://link-springer-com.ezproxy2.utwente.nl/articles/10.1186/s40294-018-0052-8https://link-springer-com.ezproxy2.utwente.nl/article/10.1186/s40294-018-0052-8`.

[53] F. Pedregosa FABIANPEDREGOSA, V. Michel, O. Grisel OLIVIERGRISEL, M. Blondel, P. Prettenhofer, R. Weiss, J. Vanderplas, D. Cournapeau, F. Pedregosa, G. Varoquaux, A. Gramfort, B. Thirion, O. Grisel, V. Dubourg, A. Passos, M. Brucher, M. Perrot andÉdouardand, a. Duchesnay, and F. Duchesnay EDOUARDDUCHESNAY. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. ISSN 1533-7928. URL `http://jmlr.org/papers/v12/pedregosa11a.html`.

[54] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 12 2007. ISSN 07421222. doi: 10.2753/MIS0742-1222240302.

[55] J. L. Peterson. Petri Nets*. *Computing Surveys*, 9(3):223–252, 1977.

[56] Process Mining Group of the Fraunhofer Institute for Applied Information Technology. pm4py - Process Mining for Python. URL `https://pm4py.fit.fraunhofer.de/docs`.

[57] Project Mesa Team. Mesa Documentation, 2023. URL `https://mesa.readthedocs.io/en/stable/overview.html`.

[58] RapidMiner. RapidMiner Platform. URL `https://rapidminer.com/platform/`.

[59] Z. Reitermanová. Data Splitting. In *WDS*, pages 31–36, 2010. ISBN 9788073781392.

[60] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*, pages 25–34, 8 1987. doi: 10.1145/37401.37406. URL `https://dl-acm-org.ezproxy2.utwente.nl/doi/10.1145/37401.37406`.

[61] S. Robinson. Simulation Model Verification And Validation: Increasing The Users' Confidence . In *Proceedings of the 1997 Winter Simulation Conferene*, pages 53–59, 1997.

[62] R. Rosés Brüngger, C. Kadar, and I. P. Cvijikj. Design of an Agent-Based Model to Predict Crime (WIP). In *Summer Computer Simulation Conference*, pages 1–6, 2016. doi: 10.5555/3015574.3015629.

[63] A. Rozinat and W. M. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1):64–95, 3 2008. ISSN 0306-4379. doi: 10.1016/J.IS.2007.07.001.

[64] A. V. Sandita and C. I. Popirlan. Developing A Multi-Agent System in JADE for Information Management in Educational Competence Domains. *Procedia Economics and Finance*, 23:478–486, 2015. ISSN 2212-5671. doi: https://doi.org/10.1016/S2212-5671(15)00404-9. URL `https://www.sciencedirect.com/science/article/pii/S2212567115004049`.

[65] R. G. Sargent. Verification and validation of simulation models. *Proceedings - Winter Simulation Conference*, pages 166–183, 2010. ISSN 08917736. doi: 10.1109/WSC.2010.5679166.

[66] T. C. Schelling. Micro Motives and Macro Behavior. *Fels Lectures on Public Policy Analysis*, page

252, 1978. URL https://books.google.com/books/about/Micromotives_and_Macrobehavior.html?hl=nl&id=4C5mQgAACAAJ.

[67] C. Schröer, F. Kruse, and J. M. Gómez. A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science*, 181:526–534, 1 2021. ISSN 1877-0509. doi: 10.1016/J.PROCS.2021.01.199.

[68] Scikit Learn. sklearn.tree.DecisionTreeClassifier — scikit-learn 1.2.2 documentation, 3 2023. URL https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier.predict_proba.

[69] R. Shannon. *Systems simulation : the art and science*. Prentice-Hall, 1975. ISBN 0138818398. URL https://books.google.com/books/about/Systems_Simulation.html?hl=nl&id=cWpRAAAAMAAJ.

[70] P. O. Siebers, C. M. MacAl, J. Garnett, D. Buxton, and M. Pidd. Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, 4(3):204–210, 9 2010. ISSN 17477786. doi: 10.1057/JOS.2010.14/TABLES/1. URL https://link.springer.com/article/10.1057/jos.2010.14.

[71] Y.-y. Song and Y. Lu. Decision tree methods: applications for classification and prediction. *Shanghai Arch Psychiatry*, 27(2):130–135, 4 2015.

[72] R. Šperka, M. Spišák, K. Slaninová, J. Martinovič, and P. Dráždilová. Control Loop Model of Virtual Company in BPM Simulation. In *Soft Computing Models in Industrial and Environmental Applications*, volume 188, pages 515–524. Springer, 2013. doi: 10.1007/978-3-642-32922-7{_}53.

[73] F. Szimanski, C. G. Ralha, G. Wagner, and D. R. Ferreira. Improving Business Process Models with Agent-Based Simulation and Process Mining. In *Enterprise, Business-Process and Information Systems Modeling*, pages 124–138, 2013. doi: 10.1007/978-3-642-38484-4{_}10.

[74] P. Taillandier, B. Gaudou, A. Grignard, Q. N. Huynh, N. Marilleau, P. Caillou, D. Philippon, and A. Drogoul. Building, composing and experimenting complex spatial models with the GAMA platform. *GeoInformatica*, 23(2):299–322, 4 2019. ISSN 13846175. doi: 10.1007/S10707-018-00339-6/FIGURES/6. URL https://link-springer-com.ezproxy2.utwente.nl/article/10.1007/s10707-018-00339-6.

[75] The Anylogic Company. Agent-based modeling | AnyLogic Help. URL https://anylogic.help/anylogic/agentbased/index.html.

[76] The pandas development team. pandas-dev/pandas: Pandas, 5 2023. URL https://zenodo.org/record/7979740.

[77] S. Tisue, U. Wilensky, and S. ; • Tisue. NetLogo: Design and Implementation of a Multi-Agent Modeling Environment. In *Agent*, 2004. URL https://ccl.northwestern.edu/papers/agent2004.pdf.

[78] D. S. Utomo, B. S. Onggo, and S. Eldridge. Applications of agent-based modelling and simulation in the agri-food supply chains. *European Journal of Operational Research*, 269(3):794–805, 9 2018. ISSN 0377-2217. doi: 10.1016/J.EJOR.2017.10.041.

[79] W. van der Aalst. Process Mining. *ACM Transactions on Management Information Systems*, 3(2): 1–17, 7 2012. ISSN 2158-656X. doi: 10.1145/2229156.2229157.

[80] W. Van Der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 9 2004. ISSN 10414347. doi: 10.1109/TKDE.2004.47.

[81] W. van der Aalst, A. Adriansyah, A. K. A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. van den Brand, R. Brandtjen, J. Buijs, A. Burattin, J. Carmona, M. Castellanos, J. Claes, J. Cook, N. Costantini, F. Curbera, E. Damiani, M. de Leoni, P. Delias, B. F. van Dongen, M. Dumas, S. Dustdar, D. Fahland, D. R. Ferreira, W. Gaaloul, F. van Geffen, S. Goel, C. Günther, A. Guzzo, P. Harmon, A. ter Hofstede, J. Hoogland, J. E. Ingvaldsen, K. Kato, R. Kuhn, A. Kumar, M. La Rosa, F. Maggi, D. Malerba, R. S. Mans, A. Manuel, M. McCreesh, P. Mello, J. Mendling, M. Montali, H. R. Motahari-Nezhad, M. zur Muehlen, J. Munoz-Gama, L. Pontieri, J. Ribeiro, A. Rozinat, H. Seguel Pérez, R. Seguel Pérez, M. Sepúlveda, J. Sinur, P. Soffer, M. Song, A. Sperduti, G. Stilo, C. Stoel, K. Swenson, M. Talamo, W. Tan, C. Turner, J. Vanthienen, G. Varvaressos, E. Verbeek, M. Verdonk, R. Vigo, J. Wang, B. Weber, M. Weidlich, T. Weijters, L. Wen, M. Westergaard, and M. Wynn. Process Mining Manifesto. In D. F, editor, *BPM 2011 Workshops*, pages 169–194, 2011.

doi: 10.1007/978-3-642-28108-2{_}19.

[82] B. van Dongen, A. de Medeiros, H. Verbeek, A. Weijters, and W. van der Aalst. The ProM Framework: A New Era in Process Mining Tool Support. In G. Ciardo and P. Darondeau, editors, *Applications and Theory of Petri Nets*, pages 444–454, Miami, USA, 2005. 26th International Conference, ICATPN. ISBN 3540263012. URL `www.cs.fiu.edu/atpn2005`.

[83] G. van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, 3 2009. ISBN 978-1-4414-1269-0. URL `https://dl-acm-org.ezproxy2.utwente.nl/doi/book/10.5555/1593511`.

[84] D. Vymětal and R. Šperka. MAREA – From an Agent Simulation Application to the Social Network Analysis. *Procedia Computer Science*, 35(C):1416–1425, 1 2014. ISSN 1877-0509. doi: 10.1016/J.PROCS.2014.08.198.

[85] A. Weijters, W. M. P. Aalst van der, and A. K. Alves De Medeiros. *Process mining with the HeuristicsMiner algorithm*. BETA publicatie : working papers. Technische Universiteit Eindhoven, 2006. ISBN 978-90-386-0813-6.

[86] R. Wirth and J. Hipp. CRISP-DM: Towards a Standard Process Model for Data Mining. 2000. URL `http://www.cs.unibo.it/~montesi/CBD/Beatriz/10.1.1.198.5133.pdf`.

8 APPENDICES

8.1 Appendix A

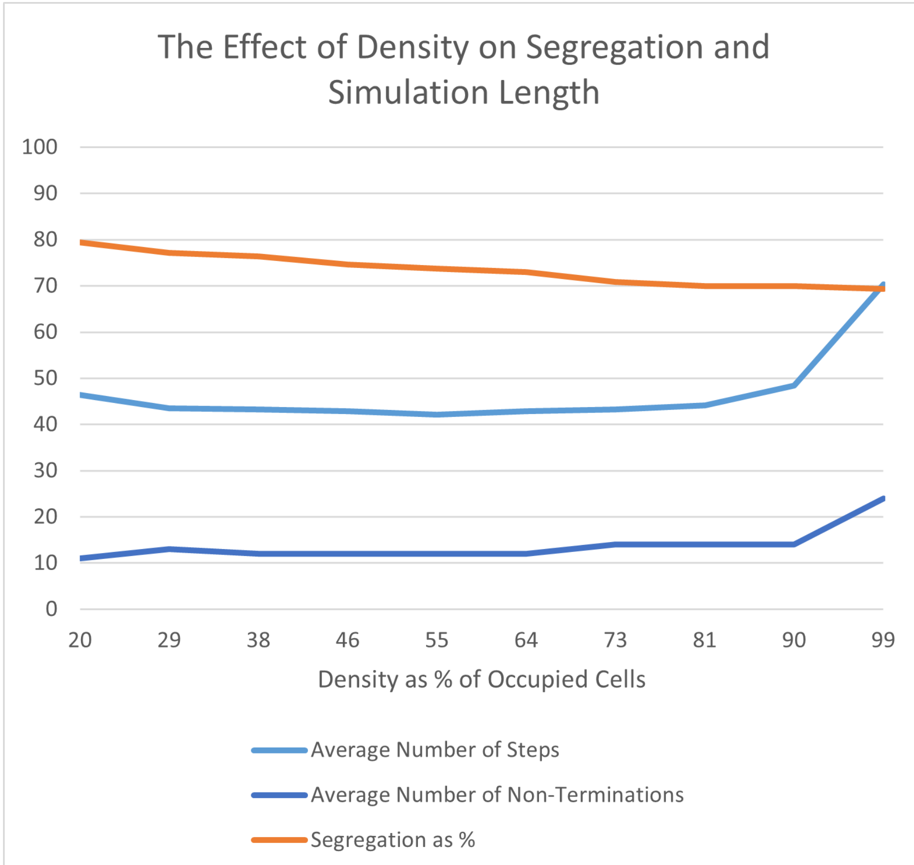

Figure 13. The effects of density on segregation in homogeneous agent experiments.

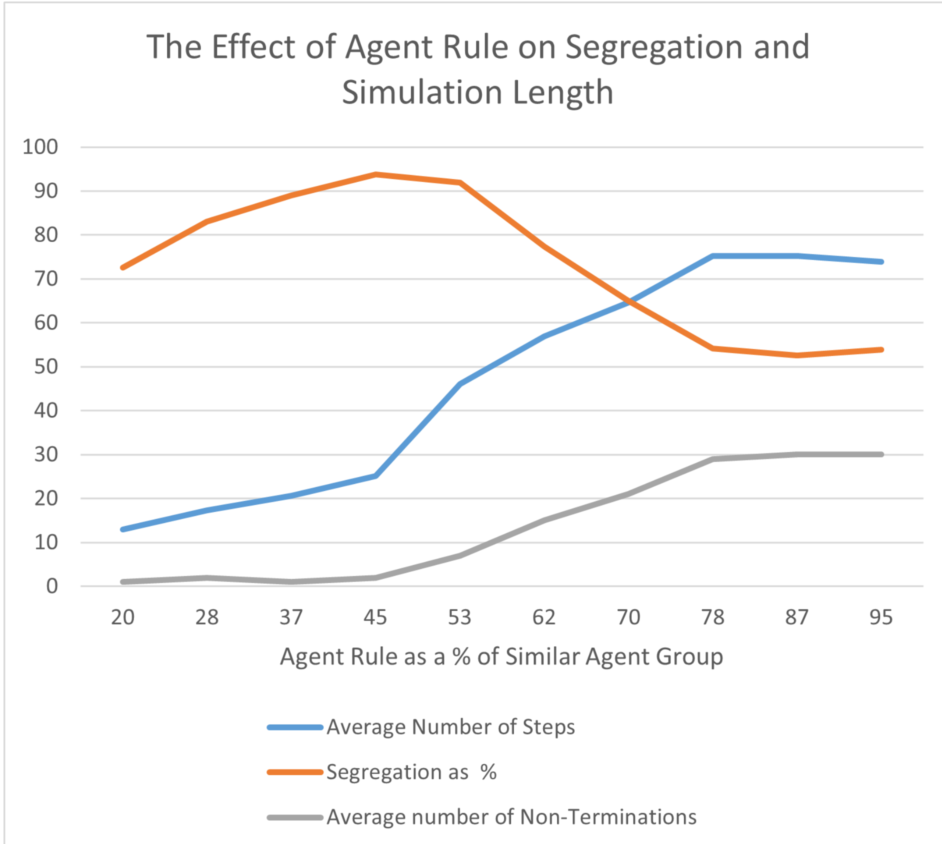

Figure 14. The effects of the agent rule on segregation in homogeneous agent experiments.

Size	Average Number of Steps	Number of Non-Terminating Runs	Segregation
20x20	46.93	138	73.18%
50x50	48.42	142	73.16%

Table 8. A comparison between different grid sizes for homogeneous experiments.

8.2 Appendix B

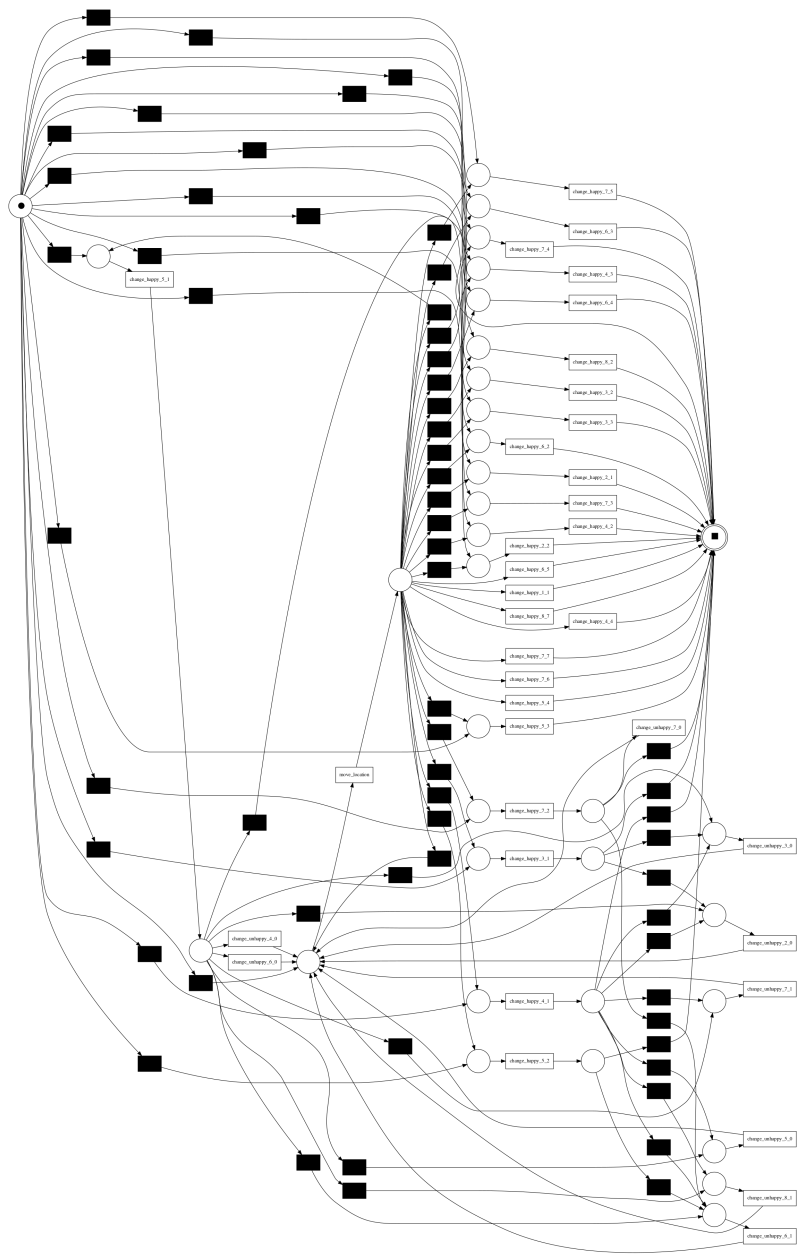

Figure 15. Generated Petri net from an experiment in the parameter search.

8.3 Appendix C

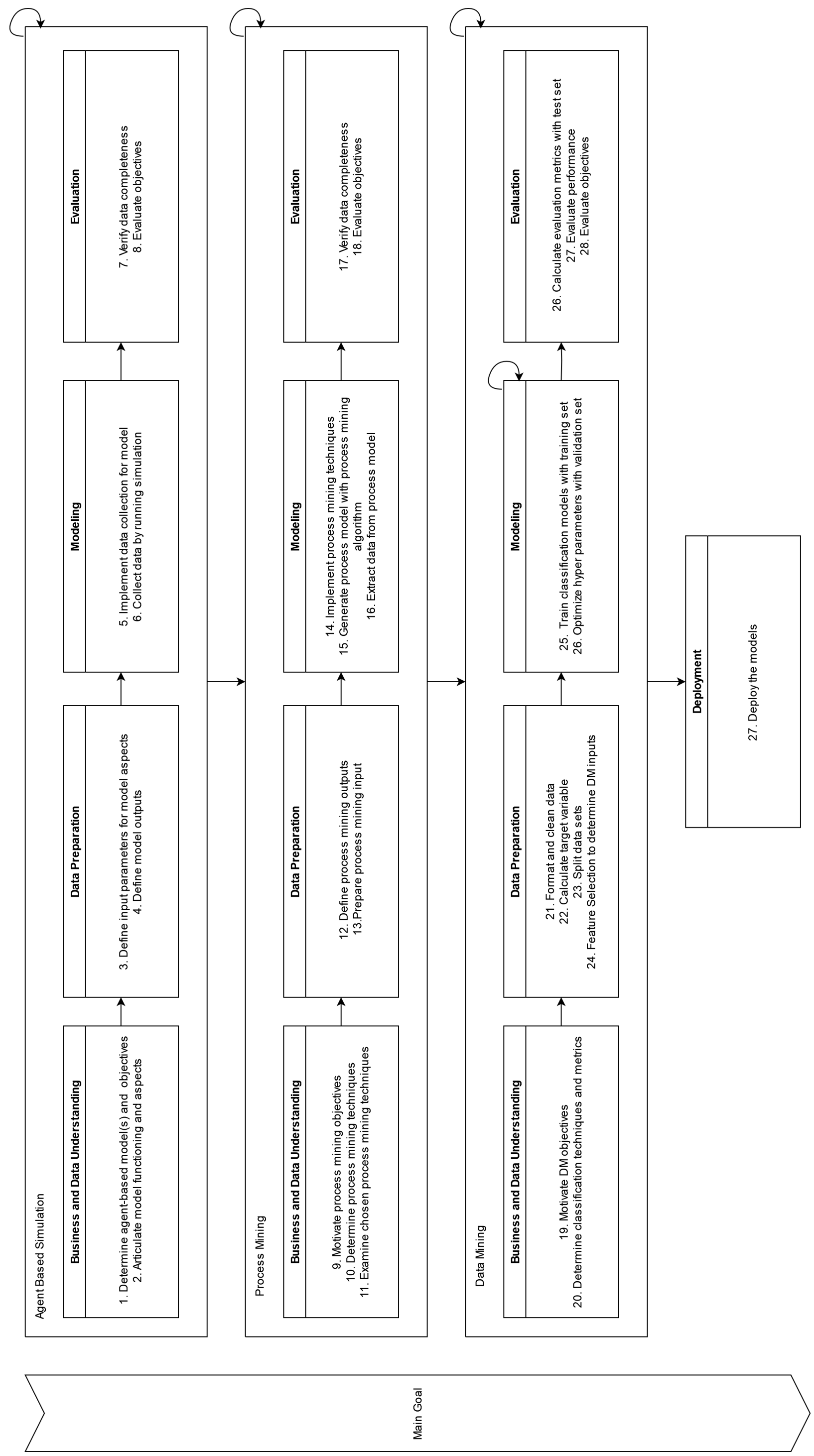

Figure 16. A larger version of Figure 7