

Author's manuscript accepted for publication in the 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW).

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

IEEE policy provides that authors are free to follow funder public access mandates to post accepted articles in repositories. When posting in a repository, the IEEE embargo period is 24 months. However, IEEE recognizes that posting requirements and embargo periods vary by funder. IEEE authors may comply with requirements to deposit their accepted manuscripts in a repository per funder requirements where the embargo is less than 24 months.

DOI: 10.1109/EDOCW.2019.00022

Cite: R. H. Bemthuis, M. Koot, M. R. K. Mes, F. A. Bukhsh, M. Iacob and N. Meratnia, "An Agent-Based Process Mining Architecture for Emergent Behavior Analysis," *2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW)*, Paris, France, 2019, pp. 54-64, doi: 10.1109/EDOCW.2019.00022.

Accepted Manuscript

An agent-based process mining architecture for emergent behavior analysis

Rob H. Bemthuis¹, Martijn Koot², Martijn R. K. Mes², Faiza A. Bukhsh³, Maria-Eugenia Iacob² and Nirvana Meratnia¹

¹*Pervasive Systems, University of Twente, Enschede, The Netherlands*

²*Industrial Engineering and Business Information Systems, University of Twente, Enschede, The Netherlands*

³*Datamanagement and Biometrics, University of Twente, Enschede, The Netherlands*

{r.h.bemthuis,m.koot,m.r.k.mes,f.a.bukhsh,m.e.iacob,n.meratnia}@utwente.nl

Abstract—Information systems leave a traceable digital footprint whenever an action is executed. Business process modelers capture these digital traces to understand the behavior of a system, and to extract actual run-time models of those business processes. Despite the omnipresence of such traces, most organizations face substantial differences between the process specifications and the actual run-time behavior. Analyzing and implementing the results of systems that model business processes tend, however, to be difficult due to the inherent complexity of the models. Moreover, the observed reality in the form of lower-level real-time events, as recorded in event logs, is seldom solely explainable by higher-level process models. In this paper, we propose an architecture to model system-wide behavior by combining process mining with a multi-agent system. Digital traces, in the form of event logs, are used to iteratively mine process models from which agents can learn. The approach is initially applied to a case study of a simplified job-shop factory in which automated guided vehicles (AGVs) carry out transportation tasks. Numerical experiments show that the workflow of a process mining model can be used to enhance the agent-based system, particularly, in analyzing bottlenecks and improving decision-making.

Keywords—Multi-agent System; Process Mining; Emergent Behavior; Enterprise Architecture; Supply Chain Logistics; Job-shop; Internet of Things

I. INTRODUCTION

Recently, emerging paradigms, such as Industry 4.0, Internet of Things (IoT), and other data-driven approaches have heightened the global need for gathering and exploiting data. The number of devices ubiquitously tracking and monitoring supply chain conditions, ranging from low-level events such as vehicle accelerations, to higher-level events such as environmental disasters, is increasing rapidly [1]. Driven by competitive forces, these colossal amounts of data (known as ‘big data’) hold the promise of supporting a wide range of supply chain functions, including procurement [2], logistics management [3], [4], and sustainability efforts [5]–[7]. As a consequence of the continuous increase in the volume and granularity of data captured by information systems [8], companies are nowadays able to monitor and capture all kinds of events affecting supply chain performance [9]. Therefore, the development of useful IT architectures and data-driven algorithms are becoming increasingly important to continuously support the companies’ need to organize their business processes in an efficient manner.

Despite the eminent need for analyzing the huge amount of data coming from today’s information systems, the vast majority of companies still face significant discrepancies between the (envisioned) process specifications and the observed reality [10]. One of the major problems in this regard is the lack of the successful integration of all subsystems into larger complex hierarchical systems with sufficient preliminary analysis, due to, for example, budget and time constraints. As a result, the system-wide impact is often not properly assessed. Another problem is that involved stakeholders typically have divergent (or even conflicting) objectives, making it challenging to find solutions to which all stakeholders are willing to commit [11]. Hence, consensus is not always reached among the players, even if supply chain coordinators and orchestration services attempt to generate commitment by providing (financial) incentives. Finally, supply chain systems may consist of many independent decision-making entities that are working in an autonomous, self-interested and not necessarily cooperative way [12]. As a result, individual players may not trust the system, may not be willing to share their information, or may not comply with the systems outcome.

The mere use of process modeling is no longer satisfactory, and transition towards integration with other fields of study such as monitoring of process execution and data analytics is, therefore, rapidly gaining attention [13]. One of the rising disciplines that has been addressing this transition is process mining. Process mining supports the analysis and understanding of the actual operational processes that are being executed by exploiting event data [14]. The diversity of supply chain players each having their own goals, own challenges with analyzing system-wide performance, and a wealth of data available, made us approach these challenges by two complementing, yet not commonly combined disciplines: process mining and multi-agent systems (MAS).

We argue there is much to be gained by using these two reference disciplines. In particular, for analyzing properties and behaviors emerging from interacting autonomous decision units, better known as emergence. First, the study of real-life enterprise data can provide valuable insights into the actual business execution and performance [15]. While the development of model-driven methods (e.g., BPMN) are often time-consuming, costly, and prone to validation

concerns, the event log analysis will only require a well-thought layout of all possible state transformations. Therefore, process mining forms an efficient method to assess the performance and discover emergent behavior of complex logistic networks. Second, a MAS is flexible, adaptable, and reconfigurable [12], which enables agents to act as a natural recourse for modeling supply chains as well. Providing insights into emergent behavior arising from decentralized decision units, such as typically modeled by agents, is one of the challenges in MAS design.

To this end, the purpose of this article is to propose an agent-based process mining architecture that will aid supply chains managers in decision-making for analyzing emergent behavior. We thereby focus on supply chain environments that can naturally be represented by a MAS. Typical examples of such environments are (i) distributed control of manufacturing operations and (ii) coordination and cooperation of logistic processes between independent supply chain actors. We validate our architecture by presenting and demonstrating a logistics supply chain case study. The novel contribution of this paper can be summarized as follows:

- A joint agent-based process mining architecture to evaluate agent-based decision rules.
- Validation of the architecture by means of a simulation study. This study comprises a classic case study, emerging from the field of supply chain logistics: the job-shop scheduling problem.
- An analysis of the impact of different agent-design choices by evaluating the results of the selected, frequently used process mining algorithms.
- A preliminary study of the impact of agents' intelligence on the overall system performance reflected in the agents' decision-making behavior.

The research methodology we adhere to in this study is Peffers design science methodology [16], as reflected by the structure of this paper. The problem statement and the research goal have been explained above. Section II covers a brief introduction to process mining and agent-based modeling, and addresses related work. Section III gives an account of the requirements of our agent-based process mining architecture. In Section IV, we present our reference architecture (the main design artefact). Section V presents the case study. Section VI validates the architecture (and demonstrates the proposed solution) by addressing an experimental setting, agent configurations, and numerical results from the logistics case study. Section VII gives a discussion. Finally, Section VIII concludes this paper with some pointer to future work.

II. BACKGROUND AND RELATED WORK

A. Process mining

Information systems typically record information about (business) processes in the form of event data [9]. Process

mining attempts to discover, conform, and improve real processes by retrieving valuable knowledge (e.g., process models) from the event logs [17]. Event logs, also referred to as transactional logs or audit trails [18], record events that occur and are usually stored in databases. Event logs can originate from Transaction Processing Systems (TPSs) [19], but an increasing number of ubiquitous sources, such as wireless sensor networks, social media, and IoT, provide real-time event data nowadays. They contain a large amount of raw data about how business processes are actually being executed.

Basically, an event log can be viewed as a set of traces, each containing activities executed for a particular process instance (see Fig. 1). Event records usually refer to an activity (i.e., a well-defined step in the process) and are affiliated with a particular case (i.e., process instance) [20]. Besides that, an event contains a timestamp and may describe additional data such as the resource or costs.

Process mining comprehends distilling a structured process description from a set of actual process executions [20]. Fig. 2 graphically depicts a conceptual model for process mining. Event logs are transformed into process models using algorithms and techniques originating from the data mining discipline. Process mining uncovers valuable information about the interaction with the information systems by discovering underlying processes from event logs. There are three main types of process mining techniques:

- Process discovery, which constructs a comprehensive process model reproducing the behavior observed in the log file [17].
- Conformance checking, which relates process models and recorded behavior to each other [22]. Methods and techniques can be used to analyze behaviors observed in event logs in the presence of a process model.
- Enhancement, which enriches the process model with data in the event log [17]. Different methods can be merged to change or extend the a-priori model.

It has been demonstrated that process mining is a promising approach for analyzing complex supply chains. First, because process mining discovers insights into system-wide behavior and, therefore, provides a shared overview of the higher-level processes. Second, process mining stimulates business process re-engineering efforts by generating im-

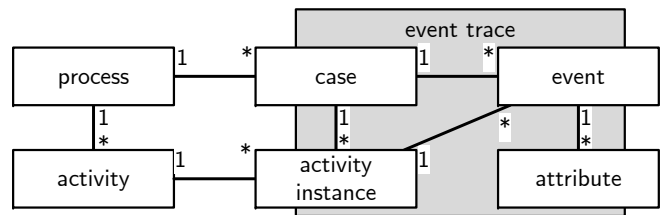


Figure 1. Connection between the process model and the event traces [21]

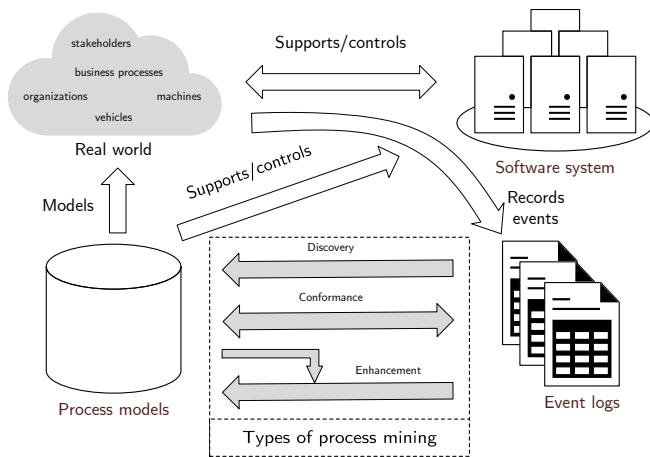


Figure 2. Conceptual process mining framework [21]

provement ideas or taking corrective actions [23]. Thus, it affects chances to improve both the individual agent's efficiency and system-wide efficiency. Third, it relies on the simple yet elegant concept of event logs, which are widely available in modern information systems [24]. By directly using log data from these systems, one can explore execution variations and intervene based on a factual understanding of the present data.

The studies presented thus far provide evidence that process mining is useful for many sectors, such as healthcare [14], [25] and education [26]. However, although process mining received enormous research interest in recent years, literature on the evaluation of cross-organizational supply chains by means of this technique is scarce.

B. Agent-based modeling

Agent-based modeling (ABM) represents a system as a collection of autonomous decision-making entities, the so-called agents. An agent is a computer system that individually assesses its situation and makes decisions autonomously based on a (simple) set of rules [27]. Agents are able to execute various tasks and provide valuable information about the dynamics of real-world systems that they emulate to fulfill their delegated objectives [28]. In addition, agents can evolve and adapt to continuously changing environments. A group of interacting agents that collectively solves complex problems is defined as a multi-agent system (MAS) [29].

An agent is characterized by its self-contained (uniquely identifiable), autonomous (acts on its own), and social behavior (interacts), which makes them suitable for modeling complex systems [30], such as supply chains. In ABM, the agent's behaviors are conditioned on its state. Hence, the set of an agent's behaviors can become richer as the set of possible states becomes richer [30]. Providing insights into emerging phenomena encapsulated by the behaviors and states is a major area of interest within the field of ABM. We refer the reader to [28] for more details about ABM.

C. Combining agent-based modeling and process mining

ABM is in particular useful when complementing it with process mining [31]. ABM mainly focuses on modeling the interaction between agents to capture the emerging behavior of complex systems, while process mining aims to discover and improve business processes (which eventually is also a way of capturing behavior of organizational systems). So, ABM can be seen as a modeling architecture while process mining could be used for analysis [31].

There are a number of reasons why these disciplines are mutually amplifying each other. First, agents can adapt and change their behaviors accordingly. Agents may update their strategies based on extracted knowledge from interaction with their environment. However, this data may be difficult to obtain, misleading or misinterpreted. Event logs, on the other hand, provide a shared knowledge source of the actual events and are not based on conjectures or intuitions. Discovered models can then act as a trusted source for the agents. Second, generated process models can be used to evaluate the emerging behavior and performance of agent-based models holistically and, therefore, can mimic the behavior of (parts of) a supply chain. Individual agents often have limited capabilities (e.g., computational power, memory, etc.) and are not able to provide this overview, but by combining the capabilities of a group of agents, process models can gradually be discovered. Third, ABM can enhance process mining by providing an additional control flow, organizational, and performance perspective. Therefore, ABM can increase the quality of constructed process models, but also vice versa. For example, an agent may detect emergent behavior that is not represented satisfactory by the discovered model. Consequently, the agent decides to generate an event for this. The discovered model is then adjusted and, in turn, other agents are informed about this update. So, ABM and process mining mutually stimulate business process improvements and are able to verify each others architectures and models.

D. Beyond the state-of-the-art

Process mining has a lot of potential, but there is still lack of articles related to a comprehensive architecture for process mining in supply chains to provide intelligence support [32]. An often faced challenge is to design algorithms that are yielding high reliability and quality of the constructed process models [33]. Furthermore, the number of publications addressing validation criteria and metrics for the quality of process models is limited [34]. By applying an agent-based architecture amplified with process mining methods, we present a novel way of assessing the quality and performance outcomes of process mining techniques.

There are relatively few studies that postulated a convergence between process mining and agent-based models. A recent publication [31] formally shows how an abstract architecture of a MAS can be analyzed by means of process

mining techniques. Authors of [35] presented a MAS that is analyzed with the use of process mining. In their work, process mining was used to validate the workflow and to analyze bottlenecks in business processes. The article [36] used process mining and agent-based simulation to perform analysis on a hospital database. Their model extracts knowledge from an existing database through simulation. Authors of [37] analyzed agent-based simulation outputs through process mining methods. In [38] an approach was presented to improve business process models based on process mining and agent-based simulation.

Collectively, these studies outline an important role of process mining in conjunction with agent-based models. However, most studies have focused on the improvement of the models discovered by means of process mining while it would be also interesting to focus on research on agent behavior, and incorporate the wealth of capabilities agents have as well. Furthermore, performance evaluation outcomes from the discovered models are often not transferred to actual knowledge as input for the agents. Moreover, an overall architecture comprehending this combination has not been proposed yet.

In this paper, we advance the state-of-the art by providing an architecture for evaluating agent-based systems by using constructed process models. This architecture enables us to not only use process mining for the discovery of processes but also to utilize the knowledge obtained from the discovered models by the agents. We validate our model by using a case study on a typical logistics problem, the so-called job-shop scheduling problem. To this end, we perform a simulation study to assess the impact of a designed agent-based model on the process models mined.

III. REQUIREMENTS

Before delving into the details of our architecture for joint process mining and agent-based modeling, we present an additional set of generic requirements with respect to the integration of both fields. Basically, the agent-based system forms a specialization of the software system represented in the process mining framework proposed by [21] in Fig. 2. While this conceptual framework is useful to demonstrate how the event logs are transformed into process models, we still need additional guidelines for the actual implementation of the required hard- and software components. Therefore, the objective of our architecture should clearly describe how the event logs are generated for the management of emergent behavior. In line with the design science methodology, we decompose the problem conceptually and infer the objectives of a solution by positioning these requirements. Based on discussions with consortium partners, as well as inspiration obtained from Industry 4.0 design principles (e.g., see [39] and [40]), we formulate a series of requirements.

Requirement 1 (Real-time data acquisition). *Changes within the agent's environment should be registered (nearly)*

real-time as traceable event data to support the agent's autonomous decision-making tasks. The context-aware information of all agents will gain insight into the overall system's performances as well.

Requirement 2 (Interoperability). *An interconnected network of sensing devices is required to enable the agents to collaborate with each other. The development of a (central) repository for traceable event data would also stimulate the agents to search actively for emergent behavior patterns.*

Requirement 3 (Modularity). *The architecture should stimulate a modular design, which is an inherent property of typical agent-based systems. Modular systems can easily be adjusted to changing conditions (e.g., emergent behavior) by replacing or expanding individual components. Agents should also be able to interact with other systems that are not explicitly modeled, since the inclusion of exogenous (system) events may be an additional valuable source of information for emergent behavior as well. Therefore a standard format (e.g., XES files) is recommended.*

Requirement 4 (Decentralized decision-making). *Agents should be capable of acting autonomously if emergent behavior is detected. Therefore, the agents should be able to explore the event logs generated by itself and other agents.*

Requirement 5 (System performance evaluation). *Since the way how agents make decisions may depend on the behavior and decisions of other agents, the overall system performance can be improved if the agents are aware of their decision's impacts. The application of process mining techniques will enable the agents to discover these overall system performances by exploiting the event logs generated.*

Requirement 6 (Strategic decision support). *The empowerment of intelligent agents with process mining techniques should support strategic modeling by taking into account emergent behavior. Instead of designing a MAS that is robust to any type of disruption, the main focus is how to adapt the system under changing circumstances. Therefore, the architecture should provide real-time understanding of decision impacts on both the agent's individual goals and the overall system's performances.*

IV. AGENT-BASED PROCESS MINING ARCHITECTURE

According to the specified requirements in the previous section, the acquisition and analysis of event logs should enable intelligent agents to act autonomously on emergent behavior. The development of such an agent-based process mining architecture requires an integrated overview of new hard- and software components. Therefore, we have designed the enterprise architecture as presented in Fig. 3, which is in compliance with the ArchiMate 3.0.1 guidelines provided by [41]. Our presented architecture enables real-time modeling of dynamically changing environments.

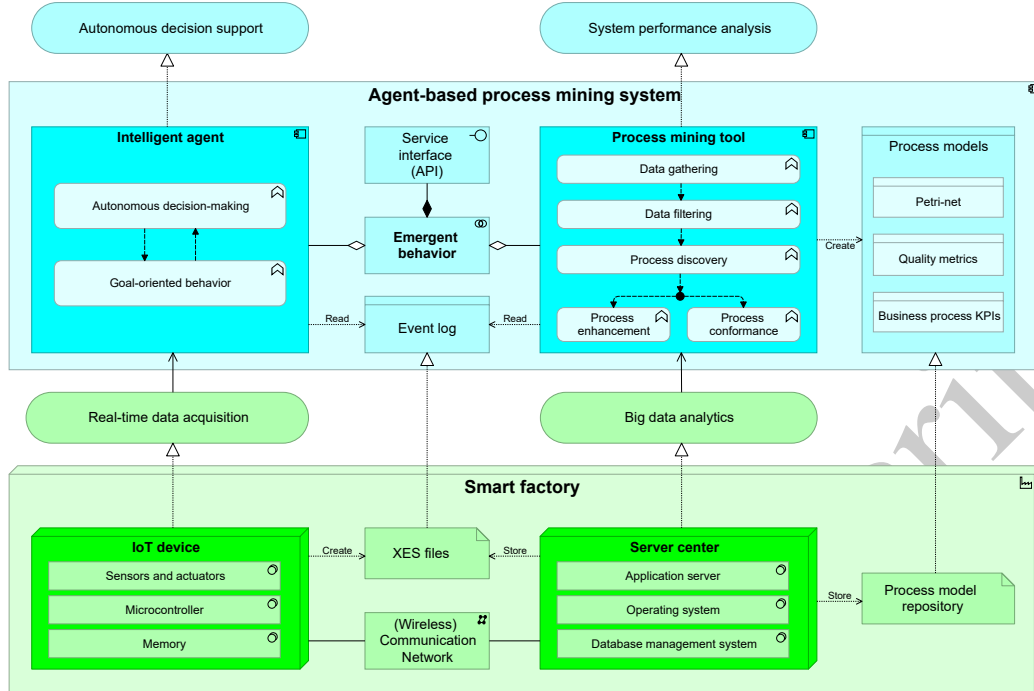


Figure 3. Agent-based process mining architecture

In our architecture, real-world systems are emulated by means of intelligent agents (e.g., computer systems or human operators). These agents represent (virtual) entities of a supply chain, such as machines and IoT devices. We empower each agent with sensing, processing, and communication capabilities to acquire (in real-time) event logs of their operations (requirement 1). The connection of these devices to a communication network facilitates agent collaboration, but also creates a repository of event logs that data-driven algorithms can exploit in their search for emergent patterns (requirement 2). Event logs are registered in XES file format, which enables agents to exchange information with each other. Therefore, it is relatively easy to expand the MAS with new types of sensing devices, as long as they can register their transactions as XES files or similar file formats (requirement 3). The architecture not only allows to capture emerging behavior from the decentralized agent-based decision units, in the form of event logs and process models, but the architecture also stimulates autonomous decision-making of the agents (requirement 4). This behavior arising from the mutual agent interaction can also be shared via service interfaces (requirement 3 and 4). In turn, process mining can assess the validity and impact of the strategies employed by the agents on the overall system (requirement 5). Thus, the architecture can be used to analyze the impact of agent intelligence and thereby evaluate the system-wide performance (requirement 6).

The resulting agent-based process mining architecture in Fig. 3 includes a hardware layer (green components) with

three main components: (1) IoT devices for the acquisition of (real-time) event logs, (2) a server center that facilitates a repository of all the agents' event logs, and (3) a communication network that enables agents to connect with each other and the database. These hardware components support the operations within the corresponding software layer (blue components), which includes two main components: (1) the intelligent agents, who will autonomously make decisions to fulfill their predefined interests and goals, and (2) the application of the process mining tool to gain insights in both the individual agent performance and system performance. This synergistic interaction between the intelligent agents and process mining tools is one of the key components of our presented architecture. The essence is that we have on the one hand, the 'virtual world of agents' represented by the intelligent agents and, on the other hand, the 'real world of process models' represented by process mining tools. An agent can update its perception based on the emergent behavior detected within the process models created and, as a response, adjust its behavior accordingly.

V. LOGISTICS CASE STUDY

This section presents an application of the presented architecture in a practical supply chain logistics case study. This case study demonstrates the efficacy of the architecture to solve a commonly known problem in the logistics field. First, we introduce the case, then describe some preliminary design choices, representing a blueprint for the experiments to be conducted.

A. Case description - simplified job-shop scheduling problem

A classical problem in the operations research literature, is the so-called job-shop scheduling problem [42]. In this problem, a set of n jobs J_1, J_2, \dots, J_n needs to be processed on m machines M_1, M_2, \dots, M_m in the smallest total makespan (i.e., the time it takes to process all jobs). Each job $j \in J$ must be processed in a given sequence of $i \in I$ operations $O_{1j}, O_{2j}, \dots, O_{ij}$, known as precedence constraints. Each operation O_{ij} is assigned to a unique machine m_{ij} and must be processed during p_{ij} non-negative units of time without any interruption. A machine can process at most one job at a time.

The consideration of this use case is motivated as follows. First, this problem is widely known in literature. The problem is known to be strongly NP-hard [42]. This is a typical situation in which detection of emergent behavior could be particularly useful for (business) process improvement goals. Second, literature about practical use cases with appropriate datasets is limited available in the field of supply chain logistics. In particular, cases where agent configurations, their performance, and their ability to learn is explicitly addressed are not available. Third, the problem context is (relatively) simple to comprehend in comparison to many large scale industrial settings. Although the classical variant has become less popular in industry, with some minor modifications to the case study, the problem will quickly become suitable to a larger set of more sophisticated variants. Fourth, the goal of this paper is not to find the best solution for a particular logistics problem instance, but rather to demonstrate the usability of our architecture.

We focus on a simplified job-shop environment in which Automated Guided Vehicles (AGVs) will operate and carry out transportation tasks from and to the machines. The job-shop factory produces three different product types and consists of four machines. Each machine is dedicated to one activity only. A different sequence of activities is required for each product type (see Fig. 4 and Table I).

A single-lane track is installed that connects the four machines with each other. The flow is bidirectional and follows a single loop configuration. In a single loop, the vehicles travel in only one loop without any shortcut or alternative routes [43]. Multiple AGVs are used for moving products between machines. The track is closed such that the

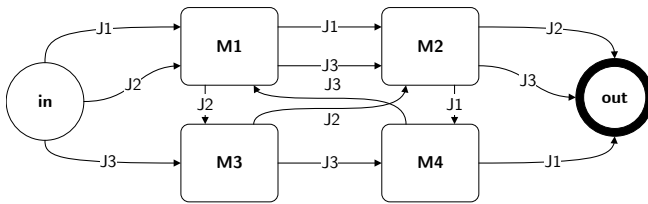


Figure 4. A simplified representation of the considered job-shop layout

Table I
JOB AND MACHINE DATA JOB-SHOP SCHEDULING PROBLEM

J	u_{1j}	u_{2j}	u_{3j}	u_{4j}	$p_{ij}, \forall i$ (minutes)
J_1	M1	M2	M4	-	$\mu = 1, \sigma = 0.15$
J_2	M1	M3	M2	-	$\mu = 1, \sigma = 0.15$
J_3	M3	M4	M1	M2	$\mu = 1, \sigma = 0.15$

outer ends are connected with each other (carousel layout, see the depiction of the track in Fig. 6). Furthermore, there is a product entrance location (M_0) and departure location (M_5). The stations are positioned sequentially next to each other near the track in this sequence: $M_0, M_1, M_2, M_5, M_3, M_4$. The distance in-between each machine in that ordering is fixed to 10 meter and M_0 is counterclockwise connected with M_4 . Thus, the total track distance is 60 meter. In this configuration, it is likely that vehicles interfere because of collision avoiding manoeuvres, resulting in interesting patterns that could emerge. Besides that, the prioritization of jobs at the machines and assignment of tasks to vehicles should be considered.

B. Architecture validation approach

Before we can test our designed agent-based process mining architecture, we specify some design choices regarding the specification of agents and agent-control rules.

In this job-shop representation, we identify three intelligent agent types: (1) machine agent, (2) AGV agent, and (3) product agent. The *machine agent* handles the processing and queuing of jobs at the machine (and entrance/exit). The *AGV agent* determines the actions carried out by the AGV. The *product agent* represents the product itself. More agent types could be defined, but considering cohesion and coupling criteria and the illustrative purpose of this case study, we will only use the aforementioned agent types. Event logs for these agents are generated as input for the process mining tool. An example is given in Table II.

Regarding the decision rules and agent relationships, we decide to focus in this study on demonstrating the interaction among one process mining tool that collects all event data from the individual software agents (machine, AGV, and product). Even though the software agents can be equipped with sophisticated self-learning capabilities, we use simple rules for each of these agents to demonstrate the feasibility of our artifact (see Section VI). The reason for not studying other configurations, is that there is a myriad number of mixed/hierarchical agent control approaches possible that can be evaluated and optimized to some degree. As said before, the purpose of this case study is not to thoroughly examine all possible agent scenarios, but to illustrate the working of our architecture. Besides that, process mining approaches and tools capable of supporting this way of experimenting is limited. Future work could consider the study of the optimal design of agent design and control rules.

VI. ARCHITECTURE VALIDATION

A. Type of simulation

To investigate the implications of alternative agent configurations and the outcome of various process mining algorithms, we use simulation. Simulation can be used to systematically evaluate a wide spectrum of model settings and to study the long-term behavior [44]. Another reason for using a simulation model is that we can strictly control the environment and we can, to some extent, verify the process models and results. That is, the mined process models can be used to verify the simulation model, but also vice versa.

More specifically, discrete-event simulation is used. This type of simulation can only change its state at a countable number of points in time [44], which is useful for event log generation. To mimic the distributed decision-making in practice, we consider agent-based simulation, which is a special form of discrete-event simulation [44]. This model is able to generate event logs. In turn, these event logs are used to establish process models.

B. Simulation setup

We will use a simulation model to evaluate multiple system configurations by adjusting the considered AGV agent's decision rules. We filter the event logs to discover a process model (Petri-nets), including some performance/quality metrics. This experimental approach is visualized in Fig. 5. We conclude this section by giving an account of the numerical results.

C. Agent-based planning and control scenarios

Two different agent-control rules are considered. Besides that, we vary the number of AGVs in the system, which we also consider as part of considered scenarios. The other two set of rules are related to the AGV planning and control and are similar for each AGV. We consider three scenarios per set of rules, so in total there are 27 combinations of scenarios possible:

- 1) Number of vehicles: 4; 5; and 6;
- 2) Vehicle driving direction: (1) forward; (2) backward; and (3) forward and backward;
- 3) AGV dispatching: (1) random; (2) longest waiting vehicle; and (3) nearest vehicle.

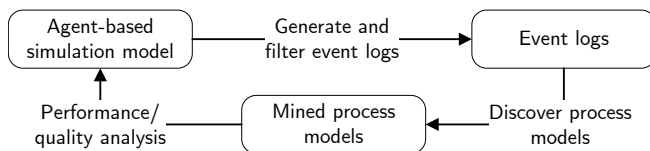


Figure 5. Experimental setting

D. Simulation model

A simulation model is developed to assess the agent performance and the system-wide performance. The model is built in 'Tecnomatix Plant Simulation' [45]. A snapshot of the simulation's graphical user interface (GUI) is shown in Fig. 6. All experiments are simulated for one complete day (24 hours), the corresponding event logs are saved in CSV format for further processing.

E. Simulation assumptions and simplifications

Note that the simulation model forms an abstract representation of the case study presented earlier in Fig. 4. Consequently, several assumptions and simplifications are required to run the simulation model properly. We recorded details of the assumptions and simplifications based on discussions during the modeling. The AGV related assumptions and simplifications are discussed first, since all experimental factors include AGV agent-control rules only:

- The AGVs always move with a speed equal to 1.0 m/s; acceleration and deceleration is excluded;
- All AGV dimensions are equal to 1 meter only;
- An AGV can only accept a new transport request:
 - if one or more products require transportation;
 - if the AGV is idle.
- An AGV will only drive:
 - if a transport request is assigned to the AGV;
 - if the vehicle dodges for another activated vehicle;
 - if the road in front of the AGV is not blocked.
- An AGV will pause:
 - if no transport request is assigned to the AGV;
 - if the AGV successfully unloads its content at the product's destination;
 - if its front road is blocked by another paused AGV.
- An AGV will always finish its transport request before it is allocated to a new job;
- AGVs will apply the same collision control:
 - a moving AGV may always push an idle AGV temporarily forward;
 - priority is randomly allocated if the collided vehicles are both moving.

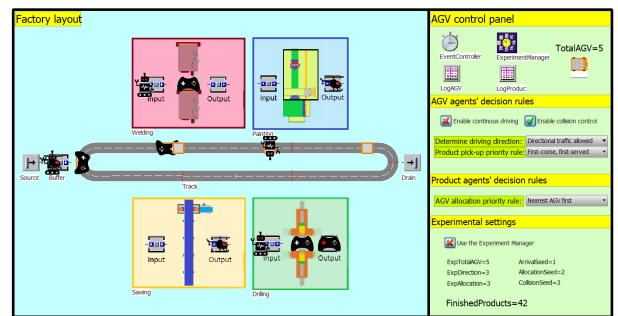


Figure 6. Graphical user interface of the simulation model

The AGV related assumptions and simplifications ensure that the AGVs can move around without any potential deadlocks. Some new levels of abstraction are also required for validation purposes:

- The factory’s track length is equal to 60 meter in total, while the departments’ input/output (I/O) points are equally distributed over the track:
 - the product entrance is located at 0m;
 - the product departure is located at 30m;
 - the I/O points of all four machines are located at 10m, 20m, 40m and 50m respectively.
- The distances between AGVs and I/O points are determined based on the front position of the vehicles;
- There is no downtime included for all resources (e.g., failures, unavailable assistance, setup, etc.);
- The processing activities of all products are ‘first-in first out’ sequenced (FIFO);
- All resources can process one product only at the same time (capacity=1);
- One physical inventory is installed for all arriving products, the capacity is unlimited;
- Two physical inventories are installed at each machine:
 - one input buffer with unlimited capacity;
 - one output buffer with unlimited capacity;
- All machines have a normally distributed processing time (in minutes) with $\mu=1.00$, $\sigma=0.15$, minimum=0.00 and maximum=2.00;
- The time between product arrivals is normally distributed (in minutes) with $\mu = 1.00$, $\sigma = 0.15$, minimum=0.00 and maximum=2.00;
- The type of product arrival (J) is uniformly distributed, resulting in a probability equal to 1/3 that any product type is selected;
- All idle products will wait for transportation/processing in one of the physical inventories installed;
- All products leave the factory at the same sink, no physical inventory is required before departure.

F. From recorded events to quality/performance metrics

The different set of rules can provide different system performances. These performances are not directly determined by the simulation model itself, neither are they known beforehand. However, the state modifications of both products and resources are registered into the corresponding event logs. Therefore, multiple experiments are conducted to obtain event logs for the alternative scenarios. Table II gives an example of such an event log. These event logs are finally exported into the process mining software ‘ProM’ to determine the system’s emergent behavior and overall performance, which were previously unknown. Our approach from events recorded in the simulation model to performance/quality metrics is as follows:

- 1) Convert the event logs from CSV to XES format:

- a) *timestamp* gives the start time of all activities;
- b) *product* groups all events into traces;
- c) *activity* describes the alternative event classifications;
- d) *life cycle* decodes the status of the event (start, complete, waiting, in progress, blocked);
- e) *resource* represents the organizational equipment required.

- 2) Filter the event log using Simple Heuristics (i.e., remove all traces that are not fully processed yet);
- 3) Construct the system’s Petri-net by allocating process discovery algorithms, based on the filtered event logs;
- 4) Replay the log on the Petri-net for performance/conformance analysis;
- 5) For selected process discovery algorithms, discover quality/performance metrics of the Petri-net.

G. Process mining algorithms

The following three frequently used process mining algorithms are considered: Alpha (α), Integer Linear Programming (ILP), and Inductive miner. Also, three commonly used quality metrics are implemented: fitness, precision, and generalization. The *fitness* of a model quantifies the fraction of the log supported by the model, *precision* quantifies the fraction not observed in the log, and *generalization* quantifies the probability that previously unseen behavior is supported by the model [46].

H. Results

Various model analyses are conducted. Table III gives an overview of the results of applying process discovery algorithms using the event logs. The conformance/performance analysis can be used to simultaneously assess the configurations simulated for potential emergent behavior. The raw output data of all simulations are also published in [47].

An example of the discovered process model of the scenario that yields the lowest average throughput time is shown in Fig. 7. We have addressed only one KPI, while the obtained process models can be evaluated on a wide variety of alternative process performance/quality indicators. For example, the Petri-net in Fig. 7 depicts the average throughput time per activity for all different product types separately. Since

Table II
AN EXCERPT OF THE PRODUCT EVENTS GENERATED BY THE SIMULATION MODEL

Timestamp	Product	Activity	Life cycle	Resource
18-04-19 00:41	ItemA:1	Move	Start	Saw.Output
18-04-19 00:41	ItemA:1	Move	Waiting	Saw.Output
18-04-19 00:41	ItemC:3	Painting	Complete	Paint.Machine
18-04-19 00:41	ItemC:3	Move	Start	Paint.Output
18-04-19 00:41	ItemC:3	Move	Waiting	Paint.Output
18-04-19 00:42	ItemB:4	Move	In progress	AGV:3
18-04-19 00:43	ItemB:4	Move	Complete	AGV:3

Table III
RESULTS OF PROCESS MINING ALGORITHMS

Scenario	ALPHA			ILP			INDUCTIVE			Average TH ^b time
	Fit.	Prec.	Gen.	Fit.	Prec.	Gen.	Fit.	Prec.	Gen.	
411	1 ^a	0.22	0.78	1 ^a	0.91	0.98	1 ^a	0.8	0.98	73
412	1 ^a	0.49	0.7	1 ^a	0.91	0.9	1 ^a	0.8	0.9	212
413	1 ^a	0.26	0.67	1 ^a	0.91	0.97	1 ^a	0.8	0.97	74
421	1 ^a	0.22	0.8	0.99	0.91	1 ^a	0.99	0.8	1 ^a	20
422	1 ^a	0.22	0.68	1 ^a	0.91	0.96	1 ^a	0.8	0.96	107
423	0.68	0.22	0.79	0.99	0.91	1 ^a	0.99	0.8	1 ^a	21
431	1	0.22	0.98	1 ^a	0.93	1	0.98	0.83	1	49
432	1	0.22	0.99	1 ^a	0.93	1	0.98	0.82	1 ^a	132
433	1	0.22	0.98	1 ^a	0.93	1	1 ^a	0.84	1	39
511	0.99	0.22	0.78	0.99	0.91	1 ^a	0.99	0.8	1 ^a	18
512	1 ^a	0.22	0.78	1 ^a	0.91	1 ^a	1 ^a	0.8	1 ^a	21
513	0.99	0.22	0.76	0.98	0.91	1 ^a	0.98	0.8	1 ^a	18
521	0.99	0.22	0.76	0.99	0.91	1 ^a	0.99	0.8	1 ^a	18
522	1 ^a	0.22	0.79	0.99	0.91	1 ^a	0.99	0.8	1 ^a	19
523	0.99	0.22	0.75	0.97	0.91	1 ^a	0.97	0.8	1 ^a	18
531	1	0.23	0.98	0.99	0.93	1	1 ^a	0.84	1	21
532	1	0.22	0.99	1 ^a	0.93	1	1 ^a	0.84	1	33
533	1	0.23	0.97	0.99	0.93	1	0.99	0.84	1	21
611	0.99	0.22	0.76	0.99	0.91	1 ^a	0.99	0.8	1 ^a	18
612	0.99	0.22	0.79	0.99	0.91	1 ^a	0.99	0.8	1 ^a	18
613	0.99	0.22	0.75	0.97	0.91	1 ^a	0.97	0.8	1 ^a	17
621	0.99	0.22	0.78	0.99	0.91	1 ^a	0.99	0.8	1 ^a	17
622	1 ^a	0.22	0.77	0.99	0.91	1 ^a	0.99	0.8	1 ^a	17
623	0.98	0.22	0.75	0.97	0.91	1 ^a	0.97	0.8	1 ^a	18
631	1	0.23	0.98	0.99	0.94	1	1 ^a	0.86	1	21
632	1	0.22	0.98	1 ^a	0.92	1 ^a	1 ^a	0.86	1	21
633	1	0.22	0.97	0.96	0.9	1 ^a	0.98	0.83	1	19
Average	0.98	0.23	0.83	0.99	0.92	0.99	0.99	0.81	0.99	40

^aRounded by two decimals.

^bThroughput time expressed in minutes.

the main aim of this article is to validate the agent-based process mining architecture proposed, and not to determine the optimal multi-agent configuration, we will continue our discussion with the evaluation of the architecture's results.

VII. DISCUSSION

Although the considered agent designs and rules are not comprehensive and should be further scrutinized, there are some contributions for illustrating the purpose of our agent-based process mining architecture. The main aim of the framework is to acquire and analyze event logs to examine emergent behavior. We will now reflect on the degree to which the resulting process models can support the intelligent agents in their autonomous decision-making.

The average throughput time in Table III is a KPI that we can use to reflect the overall system performance; some interesting observations regarding emergent behavior can be

made by reflecting on this KPI already. For example, we observe that the throughput times in scenarios with more vehicles is considerable lower. Except for scenarios 421 and 423 that have a throughput time of roughly the same order as the scenarios with more vehicles included. We also observe that the AGV dispatching rule 'longest waiting vehicle first' performs poorly in comparison to the other two dispatching rules most of the times (e.g., scenario 412, 422, 432, 532). Finally, we observe deviations within the average throughput times for each scenario, which indicates that the type of emergent behavior differs for each set of decision rules allocated. Therefore, process mining may be used for identifying emergent behavior, since the variability between the configurations' KPIs might indicate that unexpected bottlenecks have occurred. It would be interesting for further research to see how this type of behavior evolves by taking into account more or other KPIs.

In general, an acceptable replay fitness is a prerequisite for performance evaluation [46]. However, properly dealing with precision is also important. The ILP-miner, for example, scores high on fitness and precision, while the Alpha algorithm scores the lowest on the three metrics. There is one experiment (423) in which the Alpha miner yields a significantly lower fitness than for the other experiments. Intelligent agents should have a wide variety of process mining algorithms and event data at their disposal (interoperability requirement), since the complex nature of a MAS results into a system that is continuously changing. Therefore, the decentralized decision-making units should be able to apply an appropriate algorithm to properly act on emergent behaviour. However, this would require a trade-off between the process models' quality/performance metrics and the algorithm's run time. It would be interesting to examine this flexible algorithm selection procedure for further research.

Considering the intelligent agents and process mining tools, it can be beneficial for an agent to selectively interact with other agents (interoperability and modularity requirement). For example, the agent may select another AGV routing priority rule based on some particular performance/quality score provided by a process mining algorithm (e.g., the average throughput times in Table III already indicated differences between the alternative scenarios). Similarly, the process mining tool may trigger an agent to modify its decision to improve the system performance. The continuous interaction between these agents provides opportunities for optimal self-learning capabilities. It would, therefore, be interesting to include machine learning techniques within the agents as well. In addition, our findings are also relevant for simulation modelers as the generated event logs by the simulation during the simulation study are a natural recourse for constructing process models. Lastly, the often visually appealing process mining models may also be used for verification and validation of simulation studies, and stimulate the usability and understandability of non-experts.

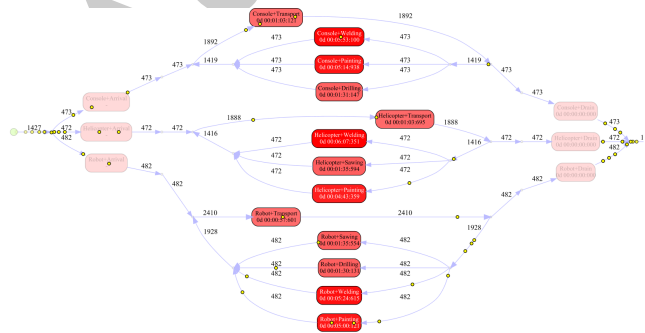


Figure 7. Inductive Visual Miner - Scenario 613 (ProM tool)

VIII. CONCLUSION AND FUTURE WORK

This study presents an approach to analyze and evaluate emergent behavior resulting from agent-based decision-making by using process mining. We introduced a holistic architecture and conducted a simulation study by solving a job-shop scheduling problem.

The preliminary findings reported in this paper provide support for the assumption that complex environments, modeled by means of agent-based systems, can be improved by exploiting the event logs. To the best of our knowledge, our study is the first to use process mining for the evaluation of emergent behavior in agent-based models. Another contribution is that we have developed an integrated overview of the required hard- and software components for the integration of both autonomous agents and process mining tools by the means of an enterprise architecture. In addition, our simulation approach for the case study in which event logs are the explicit outcome, contributes to the limited number of process mining applications available in literature. However, further (empirical) research should be conducted to validate the architecture.

The generalizability of the proposed architecture is subject to certain limitations. For instance, further modeling work will have to be conducted to employ it in large-scale (real-world) environments. Event logs often have diverse characteristics that require additional efforts to improve performance and scalability. Further research should also be undertaken to explore how the quality of mined process models is affecting the performance of agent-based models and vice versa. For instance, one could tune a particular discovery algorithm such that perturbations with respect to the representation bias are minimized. Another fruitful area for further work is to improve the agent's autonomous behavior. The goal-oriented behavior of agents combined with process mining techniques can lead to synergetic effects that ultimately improve the algorithmic design of process mining algorithms [48]. Agents can also benefit from this by equipping them with self-learning optimization capabilities. A natural progression of this work is, consequently, to automate the feedback mechanism, which may be used to analyze or affect emergent behavior in a more comprehensive manner. Lastly, although this study focuses on a single application in the field of supply chain logistics, the findings may well have a bearing on other disciplines.

ACKNOWLEDGMENT

This research is funded by the Netherlands Organization for Scientific Research (NWO) (under grant 628.009.015). The authors would also like to thank all DataRel project partners.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future

- directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [2] T. Nguyen, Z. Li, V. Spiegler, P. Ieromonachou, and Y. Lin, "Big data analytics in supply chain management: A state-of-the-art literature review," *Computers & Operations Research*, vol. 98, pp. 254–264, 2018.
- [3] N. R. Sanders, "How to use big data to drive your supply chain," *California Management Review*, vol. 58, no. 3, pp. 26–48, 2016.
- [4] D. Arunachalam, N. Kumar, and J. P. Kawalek, "Understanding big data analytics capabilities in supply chain management: Unravelling the issues, challenges and implications for practice," *Transportation Research Part E: Logistics and Transportation Review*, vol. 114, pp. 416–436, 2018.
- [5] M. H. ur Rehman, V. Chang, A. Batool, and T. Y. Wah, "Big data reduction framework for value creation in sustainable enterprises," *International Journal of Information Management*, vol. 36, no. 6, pp. 917–928, 2016.
- [6] K.-J. Wu, C.-J. Liao, M.-L. Tseng, M. K. Lim, J. Hu, and K. Tan, "Toward sustainability: using big data to explore the decisive attributes of supply chain risks and uncertainties," *Journal of Cleaner Production*, vol. 142, pp. 663–676, 2017.
- [7] R. Zhao, Y. Liu, N. Zhang, and T. Huang, "An optimization model for green supply chain management by using a big data analytic approach," *Journal of Cleaner Production*, vol. 142, pp. 1085–1097, 2017.
- [8] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of "big data" on cloud computing: Review and open research issues," *Information systems*, vol. 47, pp. 98–115, 2015.
- [9] W. M. van der Aalst, H. A. Reijers, A. J. Weijters, B. F. van Dongen, A. A. De Medeiros, M. Song, and H. Verbeek, "Business process mining: An industrial application," *Information Systems*, vol. 32, no. 5, pp. 713–732, 2007.
- [10] F. Kache and S. Seuring, "Challenges and opportunities of digital information at the intersection of big data analytics and supply chain management," *International Journal of Operations & Production Management*, vol. 37, no. 1, pp. 10–36, 2017.
- [11] W. van Heeswijk, M. Mes, and M. Schutten, "An agent-based simulation framework to evaluate urban logistics schemes," *Lecture Notes in Computer Science*, vol. 9855, pp. 369–383, 2016.
- [12] M. Mes, M. Van Der Heijden, and A. Van Harten, "Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems," *European journal of operational research*, vol. 181, no. 1, pp. 59–75, 2007.
- [13] M. Castellanos, A. A. De Medeiros, J. Mendling, B. Weber, and A. Weijters, "Business process intelligence," in *Handbook of research on business process modeling*. IGI Global, 2009, pp. 456–480.
- [14] M. Ghasemi and D. Amyot, "Process mining in healthcare: A systematised literature review," *International Journal of Electronic Healthcare*, vol. 9, no. 1, pp. 60–88, 2016.
- [15] M. Puchovsky, C. Di Ciccio, and J. Mendling, "A case study on the business benefits of automated process discovery," in *SIMPDA*, 2016, pp. 35–49.
- [16] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.

- [17] W. Van der Aalst, *Process mining: Data science in action*. Springer, 2016.
- [18] W. M. Van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. Weijters, "Workflow mining: A survey of issues and approaches," *Data & knowledge engineering*, vol. 47, no. 2, pp. 237–267, 2003.
- [19] C. Janiesch, M. Matzner, and O. Müller, "A blueprint for event-driven business activity management," in *International Conference on Business Process Management*. Springer, 2011, pp. 17–28.
- [20] W. van der Aalst and A. Weijters, "Process mining: a research agenda," *Computers in Industry*, vol. 53, no. 3, pp. 231 – 244, 2004.
- [21] 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 *et al.*, "Process mining manifesto," in *International Conference on Business Process Management*. Springer, 2011, pp. 169–194.
- [22] J. Carmona, B. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking*. Springer, 2018.
- [23] W. Van Der Aalst, "Process mining: Overview and opportunities," *ACM Transactions on Management Information Systems (TMIS)*, vol. 3, no. 2, p. 7, 2012.
- [24] F. Fox, V. R. Aggarwal, H. Whelton, and O. Johnson, "A data quality framework for process mining of electronic health record data," in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, 2018, pp. 12–21.
- [25] R. S. Mans, W. M. van der Aalst, R. J. Vanwersch, and A. J. Moleman, "Process mining in healthcare: Data challenges when answering frequently posed questions," in *Process Support and Knowledge Representation in Health Care*. Springer, 2012, pp. 140–153.
- [26] A. Bogarín, R. Cerezo, and C. Romero, "A survey on educational process mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 1, 2018.
- [27] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," in *Proceedings of the National Academy of Sciences*, vol. 99, no. 3, 2002, pp. 7280–7287.
- [28] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [29] M. Wooldridge and N. R. Jennings, "Agent theories, architectures, and languages: a survey," in *International Workshop on Agent Theories, Architectures, and Languages*. Springer, 1994, pp. 1–39.
- [30] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation," in *Proceedings of the Winter Simulation Conference, 2005*. IEEE, 2005, pp. 14–pp.
- [31] S. Ito, D. Vymětal, R. Šperka, and M. Halaška, "Process mining of a multi-agent business simulator," *Computational and Mathematical Organization Theory*, vol. 24, no. 4, pp. 500–531, 2018.
- [32] Y. Wang, F. Caron, J. Vanthienen, L. Huang, and Y. Guo, "Acquiring logistics process intelligence: Methodology and an application for a chinese bulk port," *Expert Systems with Applications*, vol. 41, no. 1, pp. 195–209, 2014.
- [33] H. C. Lau, G. T. Ho, Y. Zhao, and N. Chung, "Development of a process mining system for supporting knowledge discovery in a supply chain network," *International Journal of Production Economics*, vol. 122, no. 1, pp. 176–187, 2009.
- [34] A. Rozinat, A. K. A. de Medeiros, C. W. Günther, A. Weijters, and W. M. van der Aalst, "The need for a process mining evaluation framework in research and practice," in *International Conference on Business Process Management*. Springer, 2007, pp. 84–89.
- [35] M. Halaška and R. Šperka, "Advantages of application of process mining and agent-based systems in business domain," in *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*. Springer, 2018, pp. 177–186.
- [36] V. Augusto, X. Xie, M. Prodel, B. Jouaneton, and L. Lamarsalle, "Evaluation of discovered clinical pathways using process mining and joint agent-based discrete-event simulation," in *Proceedings of the Winter Simulation Conference, 2016*. IEEE Press, 2016, pp. 2135–2146.
- [37] 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*. Springer, 2013, pp. 515–524.
- [38] 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*. Springer, 2013, pp. 124–138.
- [39] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *2016 49th Hawaii international conference on system sciences (HICSS)*. IEEE, 2016, pp. 3928–3937.
- [40] S. Mittal, M. A. Khan, D. Romero, and T. Wuest, "Smart manufacturing: characteristics, technologies and enabling factors," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 233, no. 5, pp. 1342–1361, 2019.
- [41] A. Josey, *Archimate® 3.0. I-A pocket guide*. Van Haren, 2017. [Online]. Available: <http://pubs.opengroup.org/architecture/archimate3-doc/toc.html>
- [42] E. L. Lawler, J. K. Lenstra, A. H. R. Kan, and D. B. Shmoys, "Sequencing and scheduling: Algorithms and complexity," *Handbooks in operations research and management science*, vol. 4, pp. 445–522, 1993.
- [43] T. Le-Anh and M. De Koster, "A review of design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 171, no. 1, pp. 1–23, 2006.
- [44] A. Law, *Simulation Modeling and Analysis*, 5th ed. McGraw-Hill Education, 2015.
- [45] Siemens PLM Software, "Plant simulation," 2019. [Online]. Available: <https://www.plm.automation.siemens.com>
- [46] J. C. Buijs, B. F. van Dongen, and W. M. van der Aalst, "Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity," *International Journal of Cooperative Information Systems*, vol. 23, no. 1, 2014.
- [47] R. H. Bemthuis, M. Koot, M. R. K. Mes, F. A. Bukhsh, M.-E. Iacob, and N. Meratnia, "Data underlying the publication: An agent-based process mining architecture for emergent behavior analysis," *4TU.Centre for Research Data*, 2019, <https://doi.org/10.4121/uuid:9e430177-1dd0-40e9-b48a-8eb39124ef4c>.
- [48] M. Ghasemi and D. Amyot, "From event logs to goals: a systematic literature review of goal-oriented process mining," *Requirements Engineering*, pp. 1–27, 2019.