Agent System Mining: Vision, Benefits, and Challenges

Andrei Tour, Artem Polyvyanyy, Anna Kalenkova

Saturday 10th July, 2021

Abstract

In the Agent-Based Modeling (ABM) paradigm, an organization is a Multi-Agent System (MAS) composed of autonomous agents inducing business processes. Process Mining automates the creation, update, and analysis of explicit business process models based on event data. Process Mining techniques make simplifying assumptions about the processes discovered from data. However, actual business processes are often more complex than those restricted by Process Mining assumptions. Several Process Mining approaches relax these standard assumptions by discovering more realistic process models. These approaches can discover more realistic process models. However, these models are often difficult to visualize and, consequently, to understand. Many MASs induce processes whose behaviors become more complex with each next embraced time step, while the complexities of these MASs remain constant. Thus, the ABM paradigm can cope naturally with the increasing complexity of the discovered process models. This paper proposes Agent System Mining (ASM) and ASM Framework. ASM combines Process Mining and ABM in the Business Process Management (BPM) context to infer MAS models of operational business processes from real-world event data, while ASM Framework maps ASM activities to different phases of the MAS modeling lifecycle. The paper also discusses the benefits of using ASM and outlines challenges associated with the implementation of the ASM Framework.

1 Introduction

Business Process Management (BPM) is concerned with improving the operational performance of organizations through the BPM lifecycle [51]. BPM uses process models to understand existing as-is processes and to communicate to-be process designs. However, manual design and update of process models take significant time and effort, even for medium-sized organizations.

Process Mining automates activities involved in creating, updating, and analyzing the explicit process models based on the knowledge about the real-world operational processes extracted from current and historical event data managed by information systems [45]. Process Mining techniques use event logs as their input. These event logs are recordings of operational processes captured by the information systems. Each event in an event log has at least three attributes:

case id, timestamp, and activity. The case id attribute identifies the case (also known as a process instance) the event belongs to. The timestamp attribute indicates a point in time when the event occurred. Finally, the activity attribute refers to the activity that triggered the event. Other Process Mining *perspectives* may require events to contain additional attributes. For example, the organizational perspective requires data on resources such as people, roles, teams, or technological entities that executed the activities [39].

Process Mining techniques make several assumptions about the input event logs. Examples of the Process Mining assumptions are: each event in the log corresponds to exactly one process instance (case) and single case notion [46]; all instances of an activity for a specific case are recorded in the event log [47]; every event in the log can be related to some activity [43]; cases presented in one event log do not share resources for the activity execution [35]; and all events related to the same case are totally ordered and linked into a single control flow [43].

These assumptions allow focusing on the development of targeted Process Mining methods but also create a gap between the Process Mining event data expectations and the event logs captured from the real-world business processes [46]. Actual business processes are often more complex than those restricted by Process Mining assumptions. As a result, process models discovered by the classical Process Mining techniques do not represent the full complexity of real-world business processes.

Several Process Mining approaches relax the standard assumptions. Object-Centric Process Mining (OCPM) challenges the single case notion assumption and the existence of precisely one process instance for each event [46]. OCPM assumes that an event may relate to multiple objects corresponding to different case notions. In turn, Queue Mining (QM) relaxes the immediate resource availability assumption [36]. QM addresses situations where multiple cases compete for limited resources, process execution is delayed, and activities are completed only when resources become available. In [58], the authors relax the assumption that a business process is recorded in a single event log by proposing a framework for top-down Process Mining from multi-sourced event logs in the context of the cross-organizational business process [42].

The relaxation of the standard Process Mining assumptions leads to discovering process models that better reflect real-world processes. However, the produced process models tend to be more difficult to visualize and understand as they contain more elements and relationships. The resulting phenomenon of complex discovered models is known as the "spaghetti process models" problem [48]. The more Process Mining assumptions are relaxed, the more critical this problem becomes.

To manage the complexity of discovered models, organizations can be considered and analyzed as socio-technical systems. Socio-technical systems are systems composed of technical and social (human) components [55]. The business processes of an organization addressed as a socio-technical system can be described from two viewpoints: macro-level and micro-level. At the macro-level, end-to-end global business processes (e.g., an order-to-cash process) are performed by the organization as a whole. At the micro-level, local procedures and work instructions are specified separately for – and are carried out by – each human or technical component. Traditional Process Mining techniques produce macro-level holistic, fully connected end-to-end control flow models.

Self-organization is one of the most flexible methods for business process adaptation [21]. Self-organizing processes have parts of their macro-level flows unspecified. Consequently, the macro-level business process behavior emerges bottom-up from local distributed interactions of the system components. These interactions occur at the micro-level within the system without a macro-level centralized end-to-end control. This lack of macro-level control leads to a situation where total ordering and casual dependency of events within a self-organized process are not guaranteed. Therefore, to produce models that encode the self-organizing behaviors, Process Mining techniques must relax the single control flow assumption that posits the total order and casual dependency for all events belonging to the same process instance.

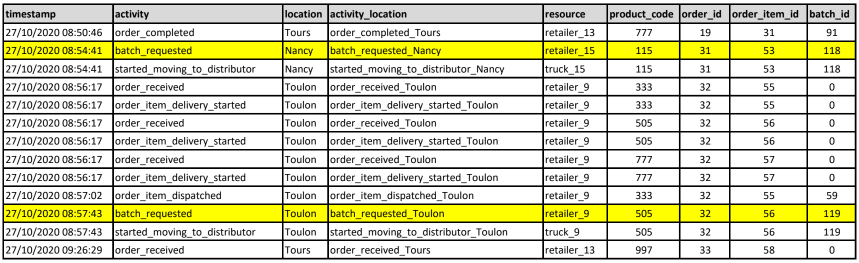
Agent-Based Modeling (ABM) is an approach for modeling and simulating organizations [4]. It can be used to conceptualize an organization as a self-organizing socio-technical system composed of autonomous agents. In ABM, observed macro-level business processes emerge from local micro-level behaviors of agents interacting with each other and the environment. This approach does not explicitly define holistic macro-level control flows. Instead, separate micro-level agent models are integrated into one Multi-Agent System (MAS) dynamically through simulated message exchange among agents. The macro-level behavior of such systems can be understood by observing and analyzing their simulation runs.

Our hypothesis is that the ABM paradigm is suitable for the automated mining of business processes from event data. This paper proposes Agent System Mining (ASM) that combines Process Mining with ABM to automate the creation of MASs that encode operational business processes of organizations. ASM supports constructing compact agent-based representations of emergent real-world business processes. This ability is based on the property of agent-based models to simulate the non-decreasing complexity of the behavior of self-organizing socio-technical systems [54]. It also provides a different perspective for analyzing processes that helps to study the macro-level business process impact of micro-level changes. The paper describes key ASM concepts, specifies ASM phases and activities, and identifies the benefits and challenges associated with ASM.

The remainder of the paper is structured as follows. Section 2 describes a motivating example identifying the existing problems and providing motivation for ASM. Section 3 introduces the key concepts and relationships ASM is based on. Section 4 establishes ASM and discusses its benefits and challenges. Section 5 reviews existing work in the related research fields. Finally, Section 6 draws conclusions and outlines directions for future work.

2 Motivating Example

To illustrate the problem of traditional Process Mining and demonstrate the potential benefits of ASM, we use a simplified example of an “order fulfillment” process performed by a hypothetical retail organization. This example uses GIS data about locations and routes in France from the “Supply Chain GIS Model” example in the AnyLogic documentation [5].



timestamp	activity	location	activity_location	resource	product_code	order_id	order_item_id	batch_id
27/10/2020 08:50:46	order_completed	Tours	order_completed_Tours	retailer_13	777	19	31	91
27/10/2020 08:54:41	batch_requested	Nancy	batch_requested_Nancy	retailer_15	115	31	53	118
27/10/2020 08:54:41	started_moving_to_distributor	Nancy	started_moving_to_distributor_Nancy	truck_15	115	31	53	118
27/10/2020 08:56:17	order_received	Toulon	order_received_Toulon	retailer_9	333	32	55	0
27/10/2020 08:56:17	order_item_delivery_started	Toulon	order_item_delivery_started_Toulon	retailer_9	333	32	55	0
27/10/2020 08:56:17	order_received	Toulon	order_received_Toulon	retailer_9	505	32	56	0
27/10/2020 08:56:17	order_item_delivery_started	Toulon	order_item_delivery_started_Toulon	retailer_9	505	32	56	0
27/10/2020 08:56:17	order_received	Toulon	order_received_Toulon	retailer_9	777	32	57	0
27/10/2020 08:56:17	order_item_delivery_started	Toulon	order_item_delivery_started_Toulon	retailer_9	777	32	57	0
27/10/2020 08:57:02	order_item_dispatched	Toulon	order_item_dispatched_Toulon	retailer_9	333	32	55	59
27/10/2020 08:57:43	batch_requested	Toulon	batch_requested_Toulon	retailer_9	505	32	56	119
27/10/2020 08:57:43	started_moving_to_distributor	Toulon	started_moving_to_distributor_Toulon	truck_9	505	32	56	119
27/10/2020 09:26:29	order_received	Tours	order_received_Tours	retailer_13	997	33	58	0

Figure 1: A fragment of an event log

The organization has multiple retailers in different locations in France and a single distributor. The retailers receive customer orders. Each order received by a retailer may have multiple order items with different product codes. To fulfill order items, the retailers request products from the distributor in batches. Vehicles move product batches from the distributor to the retailers. When a retailer runs out of stock for a specific product, they send a truck to bring a new product batch from the distributor back to the retailer. Trucks are initially co-located with retailers. When a truck receives a product batch request from a retailer, it moves to the distributor, picks up the requested product batch, and transports it back to the retailer. An order fulfillment process is completed when all items included in the order are dispatched to the customer by the retailer that received the order.

Fig. 1 shows a fragment of an event log containing event data captured from the example business process.[1] Each row in the log corresponds to one event in the order fulfillment process. The “timestamp” attribute specifies a time point when the event occurred. The “activity” attribute refers to the activity that induced the event and specifies its event type. The “location” attribute identifies the place or area where the event occurred. The “resource” attribute points to an active entity that generated the event. Finally, the “product_code”, “order_id”, “order_item_id”, and “batch_id” attributes identify, respectively, the product type, order, order item, and product batch corresponding to the event.

Fig. 2 shows a Directly-Follows Multigraph (DFM) control flow model automatically discovered using the Disco tool [18] from the example event log with the “product_code” attribute used as a case identifier. The input to the Disco

[1] The entire log can be downloaded from here: https://doi.org/10.26188/14401400.

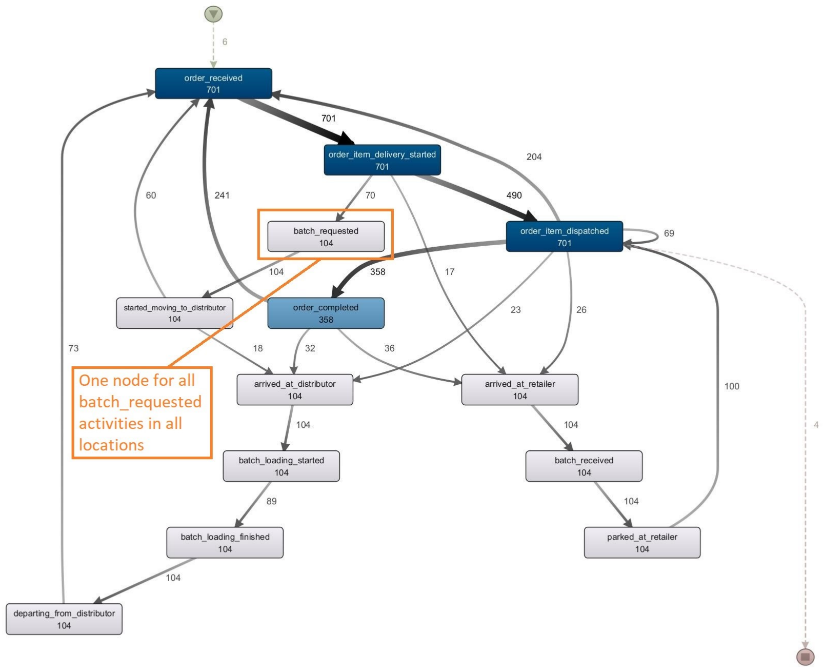


Figure 2: An order fulfillment DFM that uses location-agnostic activities

tool is a collection of traces, where each trace corresponds to a case and comprises a sequence of events ordered according to their timestamps. This control flow model is discovered under the assumption that all activities of the same type are performed in the same way without considering specific location contexts. For example, the "batch_requested" activity performed in Nancy is of the same type as the "batch_requested" activity performed in Toulon. Hence, they are represented by the same "batch_requested" node in the discovered DFM.

One can remove the assumption of location-agnostic activities and consider location-specific activities. We use pairs of activity and location attributes as a location-specific activity type to enable the discovery of a localized control flow model. For example, the "batch_requested" activity and the "Nancy" location are combined to form the location-specific activity "batch_requested_Nancy". Similarly, the "batch_requested" activity and the "Toulon" location are combined into the location-specific activity "batch_requested_Toulon".

Fig. 3 shows the DFM discovered based on location-specific activities. In this control flow model, the same activity that occurred in different locations is represented by different nodes in the DFM. For example, two instances of the "batch_requested" event that occurred in different locations, Nancy and Toulon, are represented by two separate nodes, "batch_requested_Nancy" and "batch_requested_Toulon". This location-aware version of the control flow model

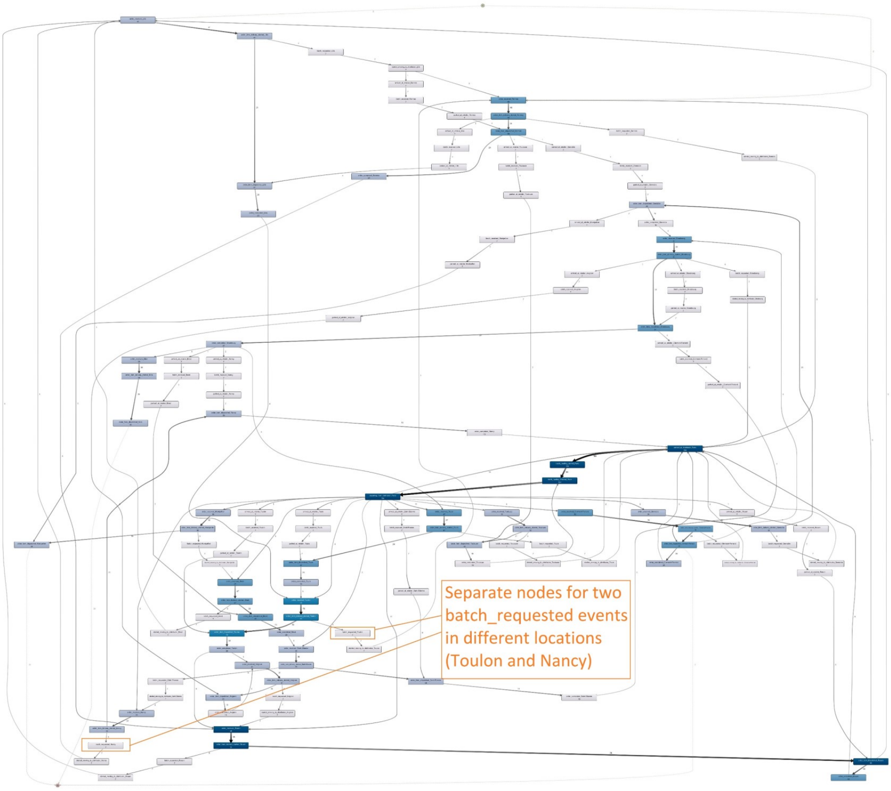
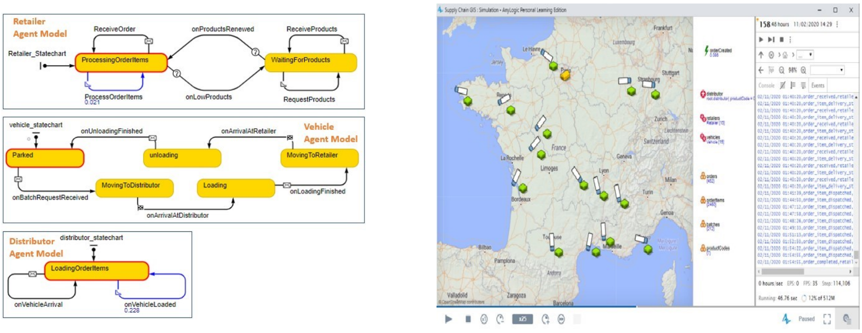

Separate nodes for two batch_requested events in different locations (Toulon and Nancy)

Figure 3: Order fulfillment example DFM with location-aware activities

is closer to the actual variability and complexity of the order fulfillment business process than the location-agnostic model in Fig. 2. However, the model in Fig. 3 is a "spaghetti process model" that is difficult to understand and use for decision-making in the BPM context.

Based on the ABM paradigm, events captured in the example event log can be interpreted as macro-level behavior emerging from micro-level behaviors of retailer agents, vehicle agents, and the distributor agent. These agents interact and act on passive objects (e.g., orders, order items, products, and product batches) in their environment (multiple locations in France). Agents can remain in the same location (e.g., retailers and distributor) or move in the environment (e.g., vehicles). Fig. 4a illustrates state chart models of micro-level behavior for the three agent types from the order fulfillment example process. The separate agent models are linked with each other and the environment model in the simulation runs by executing the behavior rules specified for each agent and the environment. The collection of all agent models, the environment model, and the interaction rules constitute the integrated MAS model. The macro-level emergent behavior is induced by simultaneous execution of all micro-level behavior rules defined for each agent and the environment included in the inte-



(a) Agent models

(b) MAS model

Figure 4: Example agent models and an integrated MAS model

grated model. Fig. 4b shows a screenshot of the example MAS model simulation run performed on the AnyLogic simulation engine [6].

The example agent models and the integrated MAS model are not “spaghetti models”; hence they are more understandable. The micro-level agent models enable a focused analysis of the local behaviors of agents. For instance, according to the retailer agent model (see Fig. 4a), a retailer does not take or process customer orders when its product stock level is low. This insight suggests an opportunity to improve the micro-level retailer behavior by making it possible for the retailer to accept customer orders while waiting for requested products from the distributor. This improvement opportunity would be difficult to identify based on the macro-level control flow models in Figures 2 and 3.

3 Organizations and Multi-Agent Systems

This section explains the key concepts of ASM, which shifts the understanding of business processes from being executed by a centralized control flow of activities to emerging as a result of interactions among people, software, and physical components within organizational systems. The remainder of this section introduces the fundamental concepts pertinent to these two viewpoints.

3.1 Organizations as Socio-Technical Systems

BPM and Process Mining often use the terms “system”, “information system”, and “business process management system” to refer to a collection of software and hardware components that capture, process, store, and produce information about different aspects of an organization and its business processes [52]. The concept of a “system” as a cohesive whole containing parts that interact to serve some purpose is used in many research disciplines and application domains to manage the complexity of a broad range of natural and artificial phenomena.

The general system theory [3] and cybernetics [53] provide the inter-disciplinary foundations for the "system" and related concepts.

The "system" concept, in the latter and arguably broader sense, can be applied to a real-world organization. The whole organization or its subset, for example, several departments or branches of the organization, can be modeled as socio-technical systems composed of social and technological components [55]. The social components include employees, teams, and departments. Software applications, robots, and equipment are examples of technological components. A socio-technical system model of an organization explicitly defines the system boundary separating the components inside the system from entities in the environment, hence outside the system. The system boundary allows identifying the system inputs and outputs used to exchange matter, energy, and information between the system and its environment.

In the order fulfillment example process described in Section 2, all the retailers, vehicles, and the distributor are components of one socio-technical system. The retailers and the distributor are social actors representing the departments of the organization. The vehicles are the technological components of the system. The system boundary is explicitly defined by enumerating all its components and their locations. For example, customers are entities outside the system that exchange information (e.g., order requests) and matter (e.g., dispatched order items) with the system.

3.2 Business Processes

A system exhibits observable behavior through changes in the system's state, inputs, and outputs over time. These changes are called *events*. From the BPM perspective, a business process is a sequence of activities manifested as events [50]. The same sequence of events produced by an organization can be interpreted as the organizational system behavior (from the socio-technical system perspective) and the organizational business processes (from the BPM perspective). By integrating the two perspectives, we can say that an organization as a socio-technical system generates its behavior by performing business processes. An organization, as a system, executes its business processes to produce the system outputs from the system inputs via state changes.

The process models in Figures 2 and 3 represent the behavior performed by the example organization as s single socio-technical system. The order delivery business process receives customer order requests as the system inputs at the retailer locations. The system performs process activities and produces dispatched order items as system outputs. Changes in the product stock levels and vehicle locations during the execution of the business process can be understood as the organizational system state changes.

3.3 Macro- and Micro-Level Behaviors

A collection of business process events can be interpreted from the whole-of-system perspective (top-down) or from the system component perspective

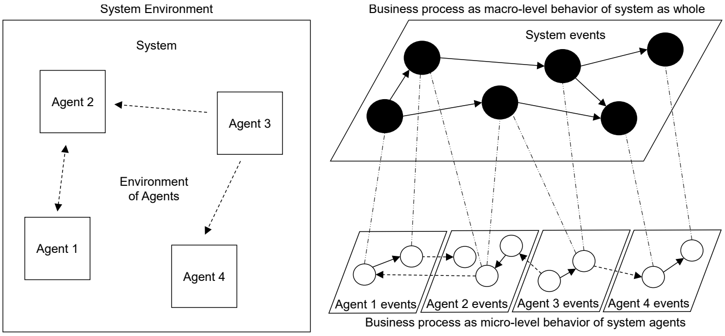


Figure 5: Macro- and micro-level abstractions of system behaviors

(bottom-up). These viewpoints correspond to the macro-level and micro-level of the system analysis. At the macro-level, the events are described as being experienced or produced by a single system-wide entity having information about all changes in the system's global state and controlling all system activities. The micro-level viewpoint is at the level of individual system components that observe and produce events in their local environments. The macro- and micro-level system viewpoints are also described in the systems modeling literature using the global/local [10] and macroscopic/microscopic [33] dichotomies. Fig. 5 illustrates abstractions of business processes as macro- and micro-level system behaviors.

The control flow models of the example order delivery process in Figures 2 and 3 are examples of macro-level behavior models. All activities represented in these models are considered as executed by the system as a whole following the global execution sequence. The local agent models in Fig. 4a are sub-models that induce the micro-level behavior, where each agent sub-model focuses on the behavior of one system component. For example, the retailer statechart describes the behavior of any retailer component included in the micro-level order delivery model.

3.4 Agents and Multi-Agent Systems

An *agent* is a central concept in the ABM paradigm. Autonomy, situatedness, proactivity, and sociality are the key aspects differentiating agents from other types of entities [56]. Autonomous agents achieve their goals by following their internal execution flows fully independent from the external control. The situatedness aspect points to the interaction of agents with their heterogeneous and dynamic environment. Proactive agents can plan and initiate their activities as opposed to passively reacting to events in the environment. Finally, social agents interact to realize shared goals or obtain necessary resources or information.

The combination of ABM and systems thinking produces a *Multi-Agent System (MAS)* concept, as a system comprised of agents [19]. The trading organization from Section 2 or any subset of its departments, teams, and assets can be modeled as a MAS. An organizational MAS can execute multiple business processes (e.g., the order delivery process). A MAS contains multiple agents (e.g., the retailers, the vehicles, and the distributor in our example) sharing the same environment. The macro-level MAS behavior is defined using the MAS inputs, outputs, states, and global behavior rules. The micro-level behaviors are defined by agent inputs, outputs, states, local behavior, and interaction rules. The environment state and behavior rules are also part of the micro-level behavior definition.

4 Agent System Mining

This section establishes ASM and defines ASM Framework. ASM automates ABM in the BPM context by extending Process Mining to analyze and discover MAS models from real-world event data. ASM Framework maps ASM activities to different phases of the MAS modeling lifecycle.

4.1 ASM Context

Fig. 6 outlines the ASM context. BPM stakeholders (Model Users) make BPM Decisions related to planning, designing, and improving Real-World Organization Business Processes using Business Process Models. These models are stored in Model Repositories. If the existing models do not meet the decision-making needs, the BPM Stakeholders may request new MAS development by providing Modeling Objectives to the MAS Modeling and Simulation Lifecycle. Modelers perform Modeling Tasks within the lifecycle phases to achieve the Modeling Objectives formulated by the BPM stakeholders. The Modeling Tasks may be performed manually or using various levels of automation. Agent System Mining (ASM) Algorithms aim to automate these Modeling Tasks by implementing ASM activities (e.g., *discover*, *check*, *enhance*, and *integrate*). These algorithms automatically produce, analyze, improve MAS models of the Real-World Organization Business Processes using the existing MAS and Other Business Process Models and Real-World and Simulated Event Logs. The resulting MAS Models are stored in Model Repositories and MAS Simulators. ASM analysis relies on Real-World Event Logs from Information Systems and Simulated Event Logs from MAS Simulators. Information Systems record Real-World Event Logs by capturing Observed Behavior of the Real-World Organization Business Processes. MAS Simulators generate Simulated Event Logs by running simulations of MAS Models.

4.2 ASM Framework

ASM Framework defines phases, tasks, activities, and artifacts involved in the development and analysis of MAS models of business processes. Fig. 7 provides an overview of the framework showing its main elements connected by logical dependency links. The diagram is inspired by the V-model lifecycle presentation

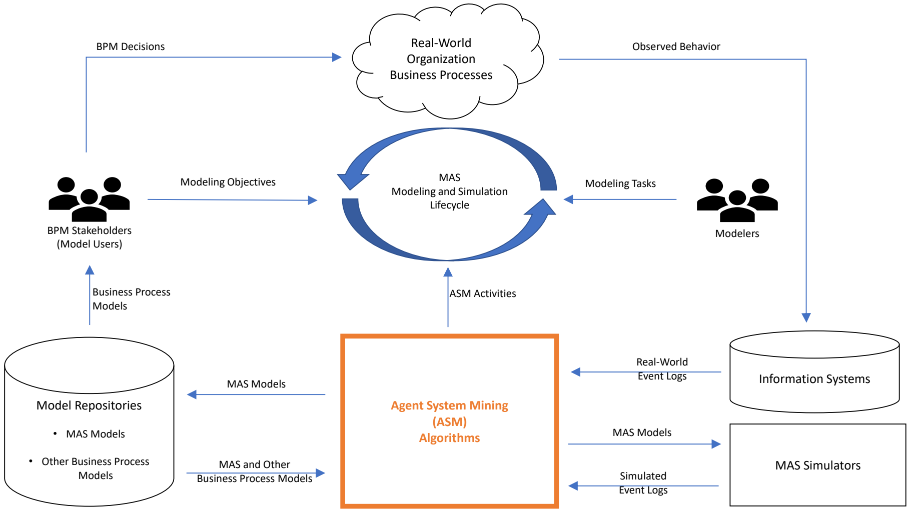


Figure 6: ASM Context

approach [34]. Systems engineers often use two-dimensional V-model diagrams to represent frameworks and lifecycles of multi-level engineered systems. In our framework, we organize the ASM Framework phases into a V-model of the MAS modeling lifecycle. The horizontal dimension from left to right represents the modeling timeline. The vertical dimension represents different levels of MAS viewpoints, including the model user viewpoint (Phases 1 and 5), the model developer viewpoints at the whole-of-model level (Phases 2 and 4), and the model developer viewpoint at the model component level (Phase 3). The framework assumes that a MAS model consists of three model component types: agent sub-models (one sub-model for each agent), a single environment sub-model, and a single interactions sub-model.

4.2.1 ASM Phases

ASM Framework defines five phases of the MAS modeling lifecycle. The phases are sequenced in time and may overlap, allowing for multiple iterations among MAS modeling tasks belonging to different ASM phases. In each phase, the modeling tasks produce one or more output ASM artifacts using the input artifacts created in the previous phases. All the tasks in Phases 1 and 5 are executed manually by the modelers or the model users. The ASM algorithms partially or fully automate the MAS activities in Phases 2, 3, and 4.

Phase 1 (Motivate) requests the development of a business process MAS model and articulates motivation for this model in terms of the BPM stakeholder needs. This phase identifies problems in the BPM context of the organization and sets MAS modeling objectives to address these problems. The objectives are formulated as benefits that are expected to be achieved by using the requested

MAS model. ASM activities are not involved in this phase.

Phase 2 (Plan) specifies the scope, required features, and constraints of the MAS model. This phase produces the following ASM artifacts: the suitable MAS metamodel that defines model concepts and their relations; the model frame that defines the context, scope, possible inputs, outputs, macro state variables for the MAS model, as well as the model requirements, assumptions, and constraints associated with the modeling objectives [57]; in-scope event logs selected from the real-world and simulated event data; in-scope models retrieved from the model repositories containing validated models previously created manually or automatically using the ASM activities. The *search* and *project* ASM activities support modeling tasks in this ASM phase.

Phase 3 (Develop) produces a MAS model containing multiple agent sub-models, one environment model, and one interactions model. The inputs to this phase include the metamodel, the model frame, and the existing event logs and models. The produced MAS model can be executed on a MAS simulator to induce the emergent macro-level behavior. This phase uses the *discover* and *enhance* ASM activities to infer the agent sub-models and the environment sub-model. The *integrate* ASM activity is involved in inferring interactions between the inferred agent and environment sub-models and integrating them into one executable MAS model.

Phase 4 (Evaluate) verifies and validates the input MAS model and its sub-models. The framework interprets the concepts of verification and validation in the same way they are defined for simulation models [32]. The model verification checks if the model and its sub-models are correct. The model validation ensures that the model is sufficiently accurate and useful for meeting the modeling objectives within its application domain. The verification and validation tasks in this phase are supported by the *diagnose* and *check* ASM activities.

Phase 5 (Use) uses the validated and verified MAS model to understand, audit, design, and improve organizational business processes within the BPM decision-making context. The BPM stakeholders and the modeling experts run simulations and perform static analysis of the model and interpret the results. The BPM insights obtained in this phase outline additional modeling needs and objectives. This leads to new MAS modeling requests and new iterations of the MAS model lifecycle starting from Phase 1. Similar to Phase 1, this phase does not use any ASM activities.

4.2.2 ASM Activities

ASM activities represent functions implemented by ASM algorithms and used in different ASM phases. ASM Framework defines seven ASM activities that partially or fully automate MAS modeling tasks involved in Phases 2, 3, and 4. Some ASM activities (e.g., *discover*, *enhance*, *check*, and *diagnose*) are inspired by the corresponding Process Mining activities [44]. Next, we detail the ASM activities.

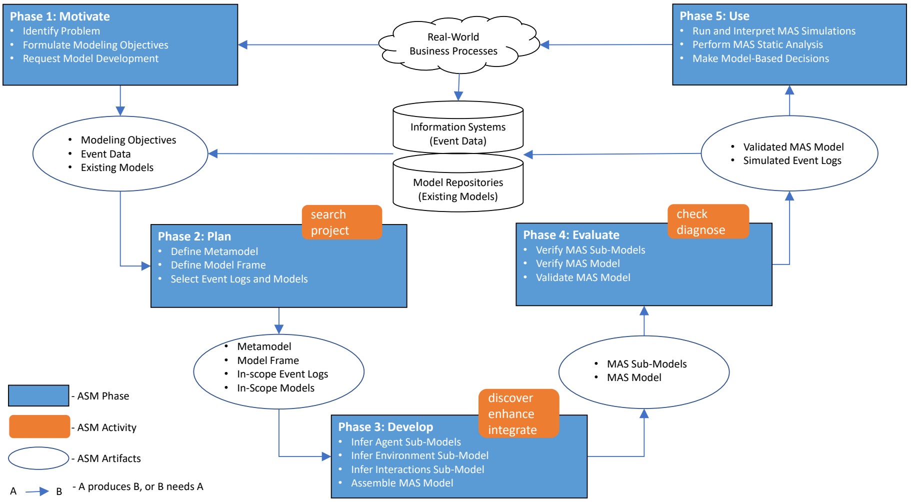


Figure 7: ASM Framework

Search activity selects a collection of existing event logs and business process models that match a given model frame. The search is performed over the organizational information systems and the model repository. This activity is used in Phase 2 of ASM Framework to identify in-scope event logs and models.

Project activity takes a model frame and existing business process models that do not match this frame and produces a MAS model that matches the given model frame. The model projection may be required when an existing model exceeds the model frame or when several existing models have to be merged to match the given model frame. This activity is used in Phase 2 to identify existing business process models that can be used in Phase 3 to develop new MAS models.

Discover activity takes the selected event log or an existing business process model as input and produces multiple agent sub-models and a single environment sub-model of a MAS model as output. An event log contains agent and environment event data and other information required for the MAS discovery. The sub-models are discovered separately for each agent and the environment.

Enhance activity takes the selected event log and an existing MAS model as input and produces enhanced versions of the input MAS agent and the environment sub-models. The enhancements may include the introduction of additional agents, state variables, or behavior rules. Similar to the *discovery* activity, the *enhance* activity does not address the problem of integration of the enhanced versions of the MAS sub-models.

Integrate activity takes the sub-models produced by the *discover* and *enhance* activities as input and produces the interactions sub-model of the MAS model as output. The produced interactions sub-model integrates the separate agent and environment sub-models.

Diagnose activity takes a MAS model as input and calculates quality indicators for the modeled system. The indicators can be calculated from the input model itself or using event logs generated by the model execution. The indicators can be defined at the whole-of-model level, and the level of individual actors or the environment.

Check activity relates the newly discovered or enhanced model versions and the existing verified and validated versions of the same model. To identify and quantify the differences, log-to-log, model-to-model, and log-to-model comparisons of real-world event logs, simulated event logs, and integrated MAS models are performed. The *diagnose* and *check* activities are used to verify and validate the inferred MAS models and their sub-models.

4.3 ASM Benefits

The benefits of ASM stem from introducing the bottom-up ABM paradigm to business process mining where, traditionally, top-down approaches dominate. We consider two categories of ASM benefits: direct and indirect. Direct ASM benefits are the benefits of automating the development of agent-based models of business processes from event logs instead of performing this modeling manually. Indirect ASM benefits are the benefits of using the ABM paradigm to model and analyze business processes.

Example direct ASM benefits are discussed below.

Evidence-based modeling. Automated analysis of event logs enables processing significantly larger amounts of data compared to manual analysis. This ability often leads to discovering additional empirical insights that otherwise remain hidden when manual process analysis techniques are used.

Shorter model development cycles. Full or partial automation of business process modeling activities shortens the time required to create and update models. Consequently, shorter model development cycles allow faster identification of changes in business processes. Thus, organizations can react faster and take timely actions to resolve arising issues.

Better conformance. By automating the comparison of existing normative process models with actual processes recorded in event logs, Process Mining reduces the time and effort of business process auditing and conformance checking, leading to better process compliance outcomes.

Direct ASM benefits can be realized by automating ASM activities applied in the context of agent-based business process management frameworks. Examples of such frameworks are agent-based BPM frameworks and approaches such as Subject-oriented Business Process Management (S-BPM) [38], Multi-Agent Business Process Modeling Notation Decision Footprint (MABPMNDF) [16], and the Knowledge Intensive Adaptive Business Process Management Framework (agileBPM) [22].

Next, we discuss example indirect ASM benefits.

Managing complex processes. The need for managing the complexity of discovered models is evidenced by the "spaghetti process models" that are difficult to understand and analyze [48]. Automatically discovered control flow models of complex business processes contain a significant number of elements and relationships, and, hence, their visual representations resemble spaghetti. In general, MAS models can induce system behaviors that over time demonstrate non-decreasing complexity [54]. In other words, a MAS model can induce system behavior whose complexity increases over time, while the size and complexity of the model stay unchanged. Consequently, MAS models can address the "spaghetti process models" problem by replacing the complex spaghetti control flow models with corresponding MAS models that induce the behavior described by the spaghetti models and are of manageable size and complexity.

Managing flexible processes. Some application domains are characterized by increased levels of business process agility and dynamism. Knowledge-intensive [22] and operational risk management processes [4] are examples of such processes exhibiting a high level of flexibility through non-linear, changing interactions among learning and adaptive participants. A suitable approach for modeling flexible business processes is encoding them as adaptive socio-technical systems [21], and ASM can support such interpretation of business processes through self-organizing MASs.

Managing context-aware processes. Context-aware models capture contextual factors inherent to the real-world business processes, for example, time, location, and socio-cultural norms [31]. MAS models can represent distributed business processes embedded into heterogeneous environments, where the environment and positions of participants in the environment are essential and not static [4]. Moreover, MAS models can explicitly capture agent mobility and changes in the process execution environment.

4.4 ASM Challenges

The key ASM objective is to achieve the highest level of automation in producing useful executable MAS models of business processes captured in event logs. ASM faces several challenges relevant to its different phases and activities on the way to this objective. Next, we discuss several important example challenges of ASM.

MAS metamodel selection. Several metamodels have been proposed for multi-agent-based simulations to achieve the ABM objectives in different application domains [23]. All agent-based metamodels introduce the notions of *agent*, *environment*, and *interaction*. However, these metamodels use different approaches for modeling details of agents and environment, micro-level and macro-level states, and interactions between the agents. The challenge, thus, is to select a MAS metamodel suitable for ASM activities in different application contexts. For example, Fig. 8 shows a simplified MAS metamodel that outlines the key agent-based concepts and relationships used in the motivating example in Section 2.

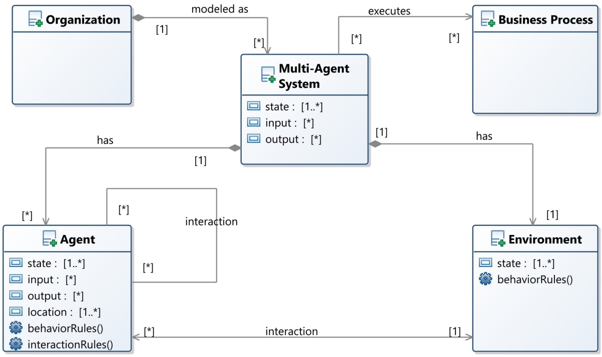


Figure 8: An example ASM metamodel

MAS model scope definition. The model scope is a frame that divides all real-world entities and events into important and not important for achieving the given modeling objectives. The defined scope is used to select event data and existing BPM and MAS models to support the ASM activities. For example, it may not be enough to use 3-dimensional space and time dimensions for the model scope definition, as multiple real-world events may happen at the same time and place. Therefore, additional criteria are required for effective scope definition (e.g., service type and customer segment). The challenge, hence, is to devise a simple MAS model scope definition method capable of producing suitable model scopes for given modeling objectives.

MAS simulation and static analysis support. MAS models obtained through the ASM activities should be suitable for execution on a simulator. Multiple simulation platforms are available for executing agent-based models. However, these platforms use different model formats and notations. Therefore, ASM integration with all available agent-based simulation platforms is not practical or feasible. At the same time, the model users and modelers should be able to perform static analysis of the discovered MAS models. Hence, on the one hand, the produced models and model components should be interpretable by expert and non-expert stakeholders. But, on the other hand, these models should be executable on multiple simulation platforms. The challenge, therefore, is to find a balance between the model readability by humans and the ability to simulate the model on multiple platforms.

MAS model validation. MAS models produced by the ASM activities must be validated to confirm that they are suitable for the intended purposes. For example, the validity of a model can be measured by comparing the event log produced by simulating the model with the event log of the corresponding real-world business process. The comparison can be performed at the micro- and macro-level system behaviors. The challenge, thus, is to identify measures and methods for the comparisons of MAS model logs and real-world business process logs. In general, other means for validating MAS models constructed from process data need to be devised.

Event data selection. An event log must contain enough information to enable ASM algorithms, for instance, to allow the identification of agents, their locations, and interactions. In addition, the data should contain information to infer the behavior rules for the agents and environment. The required data can be incomplete and distributed across several data sources. For example, in the cross-organizational context, a single complete event log of a business process may not be available due to data privacy preservation requirements [25]. Furthermore, the log data samples may be “shaped” by a specific context (e.g., day of the week, weather, and personalities of participants) and, therefore, exhibit high variability. Some available data may not be relevant to the modeling scope and objectives. All these factors contribute to the challenge of defining the methods for selecting relevant and complete event data that the ASM activities can use to produce MAS models complying with the given MAS metamodel and modeling objectives.

Agent type discovery. In a MAS model, agents can be clustered into types based on similarities in their behavior patterns and other characteristics. Consequently, all agents from the same cluster (agents of the same type) can be represented by the same agent model in the integrated MAS model. The inability to group agents based on their type can lead to complex, overfitted MAS models. Hence, the challenge is to group agents into types so that agents of the highest similarity have the same type. A solution to this challenge requires a definition of a similarity measure between agents and an approach for measuring it. Such a measure should identify agent characteristics that can be used for the comparison.

Sub-model discovery. To construct an integrated MAS model, a sub-model must be discovered for every agent type and the environment. An agent sub-model captures the agent’s reactions to the inputs, decisions, and actions. The environment sub-model encodes the environment state and describes rules for changes in the environment state. Given an event log of a business process, the challenge is to discover sub-models based on the information about recorded events that relate to multiple agent types and the environment. ASM algorithms should handle situations when one event from the log is relevant to multiple agents, the environment, a single agent, or not related to any agent within a given model frame.

Modeling language selection. Separately discovered agent and environment sub-models must be integrated into a holistic MAS model. This integration is achieved using sub-models describing agent interactions. The agent interactions sub-models should represent suitable interaction patterns, e.g., synchronous and asynchronous, with different assumptions about message delivery reliability, from the best-effort delivery to the guaranteed delivery. In addition, the integrated MAS model should be consistent and complete. A MAS model is consistent when there are no contradictions among its sub-models, and it is complete when its sub-models cover the entire scope specified by the model frame. Hence, the challenge is to identify suitable modeling languages that can describe

MAS models with a broad range of agent interaction patterns to maximize the integrated model completeness and consistency.

5 Related Work

This section provides a review of the research work related to ASM to identify reusable results and gaps in the existing knowledge base. This review follows the framework for conducting IS literature reviews proposed by vom Brocke et al [8]. It includes three steps: review the scope definition and identification of key concepts, literature search and selection, and literature analysis and synthesis.

5.1 Review Scope and Key Concepts

The review scope is defined by selecting relevant categories for the six literature review characteristics highlighted in the Cooper's taxonomy of literature reviews [11]. In this review, (i) we focus on the recent research outcomes, (ii) our goal is to summarize outcomes found in the reviewed publications related to the guiding questions, (iii) we use the key concepts to organize the search, (iv) we conduct the review from a neutral perspective without espousing our position, (v) our target audience is scholars specialized in the related fields, and (vi) we cover only literature closely related to ASM.

Fig. 9 shows a Venn diagram demonstrating research fields we identify as related to ASM, namely Agent-Based Modeling & Simulation, Business Process Management, Process Mining, and Data Mining.

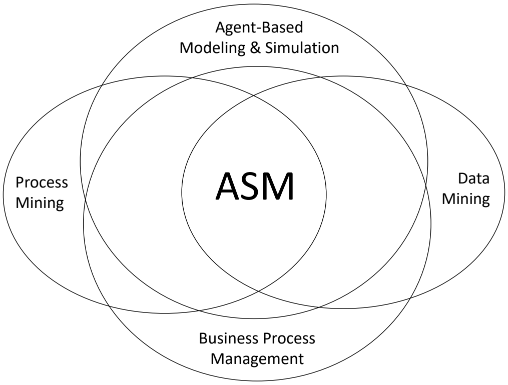


Figure 9: Research fields related to ASM

Based on the ASM purpose discussed in Section 4 and the identified related research fields, the review scope is further specified by the following guiding research questions:

RQ1 How are agent-based models used in BPM?

RQ2 Which Data Mining techniques can be used to infer (parts of) MAS models from data?

RQ3 Which Process Mining techniques can be used to infer (parts of) MAS models from event data?

The following concepts are central to the identified related research fields and the guiding questions: agent, business process, process mining, and data mining.

5.2 Literature Search and Selection

We defined the following concept-based search queries corresponding to the three guiding research questions:

Q1 agent AND “business process”;

Q2 agent AND “data mining”;

Q3 agent AND “process mining”.

The queries were configured with the following search parameters: “databases: Scopus, Web of Science”; “language: English only”; “year of publication $\geq$ 2010”; “document type: article, conference paper”; “textual content: title, abstract, keyword”; “subject area: computer science”.

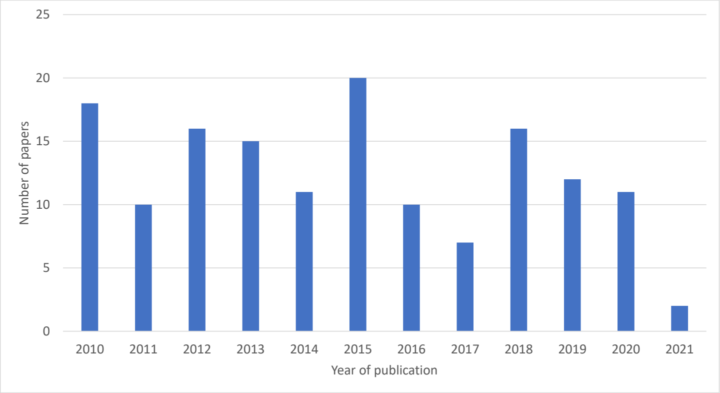


Figure 10: Number of published papers (per year)

We executed the three queries separately in the two databases on 16 February 2021. The Scopus database returned 414, 1260, and 37 results for queries Q1, Q2, and Q3, respectively. The Web of Science database returned 228, 476, and 24 results for queries Q1, Q2, and Q3, respectively. The six result sets were merged and duplicate entities removed. The merged set of results contained 2,094 publications. These publications were further filtered in two steps. First, 1563 papers were excluded based on the title relevance. In addition, 384 irrelevant papers were identified based on the abstracts. The remaining 149 papers were analyzed to fulfill the purpose of this review. [2] Fig. 10 shows an overview of the distribution of the selected papers per year.

5.3 Literature Analysis

The identified 149 relevant papers can be split into two categories. The papers from the first category focus on applying ABM for managing business processes. The papers from the second category discuss algorithms for automated verification and generation of MAS models from data. The insights obtained from the former category are summaries in Section 5.3.1 to answer research question RQ1 stated in Section 5.1, while the insights from the latter category are discussed in Section 5.3.2 and answer research questions RQ2 and RQ3.

5.3.1 ABM in BPM

Several authors discuss the idea of using the ABM paradigm and, more specifically, modeling an organization as a MAS to address decentralization, flexibility, agility, and self-adaptation of business processes in several application contexts. The multi-agent models are applied for measuring change management capability performance in a manufacturing company [1]. An agent-based simulation is used to test improvements in the flexibility and agility of business processes [17]. Several papers describe the use of ABM in the multi-organizational context. Agent-based simulation can be used to analyze cross-organizational performance [13] and is suitable for formalizing business processes in virtual enterprises [15]. Risk-aware business process management can benefit from the ability of ABM to describe agent-environment interactions [40]. MAS models are also proposed as a method for explicit representation of responsibilities and accountabilities in business processes [2], as well as for validating business requirements [27]. Finally, the concept of multi-agent cooperation is used for simulating the business processes of service businesses [41].

Multiple modeling methods, frameworks, metamodels, and formalisms are proposed for defining and implementing MAS models of business processes to support BPM activities. Subject-oriented Business Process Management (S-BPM) represents a process as a network of distributed and independent agents exchanging messages to coordinate work [30]. The agileBPM framework defines a modeling methodology to express business interest, environment, and processes according to the agent-based paradigm [22]. Hunka and van Kervel discuss how Design Engineering Methodology for Organization (DEMO) can strengthen the theoretical foundations of the Resource-Event-Agent (REA) ontology to create more precise descriptions of an organization [20]. The Input-Process-Output (IPO) abstraction [24] enables a simpler and faster approach to model a MAS in comparison with some other agent-based methodologies like Gaia [56], Tropos [7], and Multiagent Systems Engineering (MaSE) [12]. The Belief/Desire/Intentions (BDI) metamodel is used to describe a MAS behavior

[2] The complete list of selected papers can be downloaded from here: https://doi.org/10.26188/14708772.

matching that of the input real-world organization [49]. Finally, nested Petri nets (NP-nets) are used to model agents as processes and synchronize these agents into a formal MAS model of a trading software system [9].

A significant number of use cases for ABM in BPM indicates the relevance of ASM. Even though multiple frameworks and metamodels exist for MAS modeling and development, little research has been done to understand which frameworks and metamodels are suitable for inferring MAS models from business process data.

5.3.2 Algorithms for inferring MAS models from data

Most research that integrates the ABM paradigm with the Data Mining and Process Mining fields is dedicated to MAS implementations of Data Mining and Process Mining platforms and traditional Data Mining and Process Mining techniques to analyze engineered software and cyber-physical MASs. Relatively little research has been done on algorithms that generate MAS models of real-world business processes from process data.

The algorithm presented in [14] can discover micro-level agent models and link them to the input macro-level business process model. This algorithm is based on the hierarchical Markov model. It can be considered for implementation of the *enhance* ASM activity. The framework for solving the probabilistic goal recognition problem presented in [29] can be used to discover models of rational and irrational behaviors of agents. The obtained models can be used for modeling autonomous aspects of agent behavior. Data Mining methods can be used to mine context models in multi-agent interactions [37]. These methods can be reused to implement the *discover* and *enhance* ASM activities to construct context-aware environment sub-models of MASs. Mahdi and Lotfi propose algorithms for discovering agent interaction protocols and organizational structures in business processes [26]. The agent interaction models discovered using the proposed algorithms should be augmented with agent-environment interactions and individual agent behavior sub-models to fully represent agents in MAS models. Finally, the alpha-algorithm for process discovery has been used to mine a Petri net model of an individual agent in a robotic MAS [28]. The algorithm takes an event log of one robotic agent and produces a Petri net model of that agent. This approach can be used as part of an ASM Framework implementation for constructing sub-models of individual agents.

While the existing works provide useful ideas and techniques for mining parts of MAS models, they do not allow, neither individually nor collectively, implementing the ASM framework described in Section 4.2.

6 Conclusion

This paper presents a vision of Agent System Mining (ASM) as an extension of Process Mining grounded in the Agent-Based Modeling paradigm. ASM interprets business process data from an agent-based micro-level perspective. From this perspective, a business process is implicitly induced by interactions of multiple autonomous distributed agents without an explicit definition of a macro-level control flow model. As a motivation for ASM, we provide an example of mining an order delivery business process, demonstrating how Multi-Agent System (MAS) models can address the problem of "spaghetti process models" related to visualizing and understanding complex macro-level control flow models generated by traditional Process Mining techniques. To position ASM in the MAS modeling lifecycle, we introduce ASM Framework that maps ASM activities and artifacts to MAS modeling lifecycle phases and tasks. In addition, we discuss ASM benefits and challenges related to the implementation of the ASM activities.

The future research in ASM can be organized around three areas: metamodels and formalisms suitable for representing MAS models discovered from business process data, ASM algorithms for discovering and enhancing executable MAS models of business processes, and techniques for assessing the quality of MAS models discovered by the ASM algorithms. Further analysis of ASM benefits for different application domains and industry sectors will validate ASM Framework and inform additional research directions.

Acknowledgements. This research was supported by the Australian Commonwealth Government through an "Australian Government Research Training Program Scholarship."

References

[1] M.B. Ayhan, M.E. Aydin, and E. Öztemel. A multi-agent based approach for change management in manufacturing enterprises. *Journal of Intelligent Manufacturing*, 26(5):975–988, 2015.

[2] M. Baldoni, C. Baroglio, O. Boissier, R. Micalizio, and S. Tedeschi. Engineering multiagent organizations by accountability and responsibility. In *CEUR Workshop Proceedings*, volume 2495, pages 12–23. CEUR-WS, 2019.

[3] Ludwig Von Bertalanffy. An outline of general system theory. *The British Journal for the Philosophy of Science*, I(2):134–165, 1950.

[4] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002.

[5] Andrei Borshchev. Multi-method modelling: AnyLogic. In *Discrete-Event Simulation and System Dynamics for Management Decision Making*, chapter 12, pages 248–279. John Wiley & Sons, Ltd, 2014.

[6] Andrei Borshchev and Alexei Filippov. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. In *Proceedings of the 22nd International Conference of The System Dynamics Society*, pages 25–29, Oxford, England, 2004.

[7] Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, and John Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, 2004.

[8] Jan Brocke, Alexander Simons, Bjoern Niehaves, Bjorn Niehaves, Kai Reimer, Ralf Plattfaut, and Anne Cleven. Reconstructing the giant: On the importance of rigour in documenting the literautre search process. In *ECIS 2009 Proceedings*, 2009. Art. no.: 161.

[9] Julio C. Carrasquel, Irina A. Lomazova, and Iosif L. Itkin. Towards a formal modelling of order-driven trading systems using Petri nets: A multi-agent approach. In *Proceedings of the MACSPro Workshop 2019*, volume 2478, pages 92–103, Vienna, Austria, March 2019. CEUR-WS.

[10] Fritz Colonius and Wolfgang Kliemann. Examples for global behavior. In *The Dynamics of Control*, pages 353–375. Birkhäuser, Boston, MA, 2000.

[11] Harris M. Cooper. Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society*, 1(1), 1988. Art. no.: 104.

[12] Scott A. DeLoach. The MaSE methodology. In *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, pages 107–125. Springer US, Boston, 2004.

[13] A.C. Fauzan, R. Sarno, and I. MacHfud. Simulation of agent-based and discrete event for analyzing multi organizational performance. In *Proceedings - 2019 International Seminar on Application for Technology of Information and Communication*, pages 224–229. IEEE, 2019.

[14] Diogo R. Ferreira, Fernando Szimanski, and Célia Ghedini Ralha. A hierarchical Markov model to understand the behaviour of agents in business processes. In *Business Process Management Workshops*, pages 150–161, Berlin, Heidelberg, 2013. Springer.

[15] E. Garcia, A. Giret, and V. Botti. Designing normative open virtual enterprises. *Enterprise Information Systems*, 10(3):303–324, 2016.

[16] Riadh Ghlala, Zahra Kodia Aouina, and Lamjed Ben Said. Multi-agent BPMN decision footprint. In *Agent and Multi-Agent Systems: Technology and Applications*, pages 234–244, Cham, 2017. Springer International Publishing.

[17] Yiwei Gong and Marijn Janssen. Agent-based simulation for evaluating flexible and agile business processes: Separating knowledge rules, process rules and information resources. In *Lecture Notes in Business Information Processing*, pages 41–58, Berlin, Heidelberg, 2010. Springer.

[18] Christian W. Günther and A. Rozinat. Disco: Discover your processes. In *Proceedings of the Demonstration Track of the 10th International Conference on Business Process Management*, volume 940, pages 40–44, Tallinn, Estonia, September 2012. CEUR-WS.org.

[19] Christian Hahn, Cristián Madrigal-Mora, and Klaus Fischer. A platform-independent metamodel for multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 18(2):239–266, 2009.

[20] Frantisek Hunka and Steven van Kervel. A generic DEMO model for co-creation and co-production as a basis for a truthful and appropriate REA model representation. In *Lecture Notes in Business Information Processing*, volume 361, pages 203–218. Springer International Publishing AG, 2019.

[21] Christa Illibauer and Christine Natschläger. Towards flexible business processes by supporting self-organizing groups. In *Innovations in Enterprise Information Systems Management and Engineering*, pages 57–65, Cham, 2017. Springer International Publishing.

[22] Huseyin Kir and Nadia Erdogan. A knowledge-intensive adaptive business process management framework. *Information Systems*, 95, 2021. Art. no.: 101639.

[23] Franziska Klügl and Paul Davidsson. AMASON: Abstract meta-model for agent-based simulation. In *Multiagent System Technologies*, pages 101–114, Berlin, Heidelberg, 2013. Springer.

[24] Timotius Kevin Levandi, M. M. Inggriani, and Nur Ulfa Maulidevi. Multi-agent sentiment analysis using abstraction-based methodology. In *2014 International Conference on Data and Software Engineering (ICODSE)*. Piscataway, NJ : IEEE, 2014. doi: 10.1109/ICODSE.2014.7062703.

[25] Cong Liu, Hua Duan, QingTian Zeng, MengChu Zhou, FaMing Lu, and JiuJun Cheng. Towards comprehensive support for privacy preservation cross-organization business process mining. *IEEE Transactions on Services Computing*, 12(4):639–653, 2019.

[26] A. Mahdi and B. Lotfi. Discovering organizational perspective in workflow using agent approach: An illustrative case study. In *CEUR Workshop Proceedings*, volume 601, pages 84–98, 2010.

[27] N. Mitakides, N. Spanoudakis, and P. Delias. Validating business requirements using MAS analysis models. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, volume 3, pages 1693–1694. IFAAMAS, 2015.

[28] C. Ou-Yang and H. Winarjo. Petri-net integration - an approach to support multi-agent process mining. *Expert Systems with Applications*, 38(4):4039–4051, 2011.

[29] Artem Polyvyanyy, Zihang Su, Nir Lipovetzky, and Sebastian Sardina. Goal recognition using off-the-shelf process mining techniques. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1072–1080, Richland, SC, 2020. IFAAMAS.

[30] S. Raß, J. Kotremba, and R. Singer. The S-BPM architecture: A framework for multi-agent systems. In *Proceedings - IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Workshops*, volume 3, pages 78–82, 2013.

[31] Michael Rosemann, Jan Recker, and Christian Flender. Contextualisation of business processes. *International Journal of Business Process Integration and Management*, 3(1):47–60, 2008.

[32] R. G. Sargent. Verification and validation of simulation models. In *Proceedings of the 2010 Winter Simulation Conference*, pages 166–183, 2010.

[33] Hiroki Sayama. *Introduction to the Modeling and Analysis of Complex Systems*. Open SUNY Textbooks, 2015.

[34] Dieter Scheithauer and Kevin Forsberg. 4.5.3 V-model views. *INCOSE International Symposium*, 23(1):502–516, 2013.

[35] Arik Senderovich. Queue mining. In *Encyclopedia of Big Data Technologies*, pages 1351–1358. Springer International Publishing, Cham, 2019.

[36] Arik Senderovich, Matthias Weidlich, Avigdor Gal, and Avishai Mandelbaum. Queue mining – predicting delays in service processes. In *Advanced Information Systems Engineering*, pages 42–57, Cham, 2014. Springer International Publishing.

[37] E. Serrano, M. Rovatsos, and J.A. Botía. Data mining agent conversations: A qualitative approach to multiagent systems analysis. *Information Sciences*, 230:132–146, 2013.

[38] Robert Singer. Agent-based business process modeling and execution: Steps towards a compiler-virtual machine architecture. In *Proceedings of the 8th International Conference on Subject-Oriented Business Process Management*, New York, USA, 2016. Association for Computing Machinery. Art. no.: 8.

[39] Minseok Song and Wil M. P. van der Aalst. Towards comprehensive support for organizational mining. *Decision Support Systems*, 46(1):300–317, 2008.

[40] E. Sulis, I.A. Amantea, and G. Fornero. Risk-aware business process modeling: A comparison of discrete event and agent-based approaches. In *Proceedings - Winter Simulation Conference*, volume 2019-December, pages 3152–3159. Institute of Electrical and Electronics Engineers Inc., 2019.

[41] W. Tan, W. Xu, F. Yang, L. Xu, and C. Jiang. A framework for service enterprise workflow simulation with multi-agents cooperation. *Enterprise Information Systems*, 7(4):523–542, 2013.

[42] Wil M. P. van der Aalst. Configurable services in the cloud: Supporting variability while enabling cross-organizational process mining. In *On the Move to Meaningful Internet Systems: OTM 2010*, pages 8–25, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[43] Wil M. P. van der Aalst. Getting the data. In *Process Mining: Data Science in Action*, pages 125–162. Springer, Berlin, Heidelberg, second edition, 2016.

[44] Wil M. P. van der Aalst. Operational support. In *Process Mining: Data Science in Action*, pages 301–321. Springer, Berlin, Heidelberg, second edition, 2016.

[45] Wil M. P. van der Aalst. Process mining: The missing link. In *Process Mining: Data Science in Action*, pages 25–52. Springer, Berlin, Heidelberg, second edition, 2016.

[46] Wil M. P. van der Aalst. Object-centric process mining: Dealing with divergence and convergence in event data. In *Software Engineering and Formal Methods*, pages 3–25. Springer International Publishing, Cham, 2019.

[47] Wil M. P. van der Aalst and A. J. M. M. Weijters. Process mining: A research agenda. *Computers in Industry*, 53(3):231–244, 2004.

[48] Gabriel M. Veiga and Diogo R. Ferreira. Understanding spaghetti models with sequence clustering for ProM. In *Business Process Management Workshops*, pages 92–103, Berlin, Heidelberg, 2010. Springer.

[49] Y. Wautelet and M. Kolp. Business and model-driven development of BDI multi-agent systems. *Neurocomputing*, 182:304–321, 2016.

[50] Mathias Weske. Business proces modelling foundation. In *Business Process Management*, pages 71–122. Springer, Berlin, Heidelberg, 2019.

[51] Mathias Weske. *Business Process Management: Concepts, Languages, Architectures.* Springer, third edition, 2019.

[52] Mathias Weske. Evolution of enterprise systems architectures. In *Business Process Management*, pages 23–67. Springer, Berlin, Heidelberg, 2019.

[53] Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine.* The MIT Press, second edition, 1961.

[54] Stephen Wolfram. Computation theory of cellular automata. *Communications in Mathematical Physics*, 96(1):15–57, 1984.

[55] Paul Pao-Yen Wu, Clinton Fookes, Jegar Pitchforth, and Kerrie Mengersen. A framework for model integration and holistic modelling of socio-technical systems. *Decision Support Systems*, 71:14–27, 2015.

[56] Franco Zambonelli, Nicholas R. Jennings, and Michael Wooldridge. Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3):317–370, 2003.

[57] Bernard P. Zeigler, Alexandre Muzy, and Ernesto Kofman. Chapter 2 - framework for modeling and simulation. In *Theory of Modeling and Simulation (Third Edition)*, pages 27–41. Academic Press, 2019.

[58] Qingtian Zeng, Hua Duan, and Cong Liu. Top-down process mining from multi-source running logs based on refinement of Petri nets. *IEEE Access*, 8:61355–61369, 2020.