Downloaded from orbit.dtu.dk on: May 02, 2024

DTU Library

A Data Driven Agent Elicitation Pipeline for Prediction Models

Larsen, John Bruntse; Burattin, Andrea; Davis, Christopher John; Hjardem-Hansen, Rasmus; Villadsen, Jørgen

Published in:
Business Process Management Workshops

Link to article, DOI:
10.1007/978-3-030-37453-2_46

Publication date:
2020

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Larsen, J. B., Burattin, A., Davis, C. J., Hjardem-Hansen, R., & Villadsen, J. (2020). A Data Driven Agent Elicitation Pipeline for Prediction Models. In C. Di Francescomarino , R. Dijkman , & U. Zdun (Eds.), *Business Process Management Workshops* (pp. 570-582). Springer. https://doi.org/10.1007/978-3-030-37453-2_46

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A Data Driven Agent Elicitation Pipeline for Prediction Models

John Bruntse Larsen[1], Andrea Burattin[1], Christopher John Davis[1,2], MD Rasmus Hjardem-Hansen[3], and Jørgen Villadsen[1]

[1] DTU Compute, Technical University of Denmark, 2800 Kongens Lyngby, Denmark
[2] University of South Florida Saint Petersburg, United States
[3] Greve Municipality, Denmark
{jobla,andbur,jovi}@dtu.dk, davisc@mail.usf.edu

Abstract. Agent-based simulation is a method for simulating complex systems by breaking them down into autonomous interacting agents. However, to create an agent-based simulation for a real-world environment it is necessary to carefully design the agents. In this paper we demonstrate the elicitation of simulation agents from real-world event logs using process mining methods. Collection and processing of event data from a hospital emergency room setting enabled real-world event logs to be synthesized from observational and digital data and used to identify and delineate simulation agents.

Keywords: Agent-Based Simulation · Process Mining · Emergency Rooms.

1 Introduction

Emergency medicine is characterized by time and life critical episodes: the range of cases from sprained ankles to trauma patients with life threatening injuries and illnesses create a complex and dynamic working environment. This entry point to the hospital is characterized by unique and substantial challenges for resource planning and scheduling. Changes in workload can happen very quickly and bottlenecks can emerge in a matter of minutes. Our work is motivated by the aspiration of clinicians and administrators to be able to more precisely predict what the situation will look like minutes or hours from now and, ideally, understand the impact of moving staff or changing clinical priorities.

The setting for the research reported in this paper is a Joint Emergency Room (JER) in Denmark (Fælles Akutmodtagelse). In addition to major trauma and minor injuries, the JER is the entry point for patients admitted by their General Practitioner and brought by ambulance: this further increases the diversity of the symptoms and injuries presented. Nursing skills in the JER are transferable, allowing them to be readily deployed to alleviate a bottleneck. Doctors and other staff are less easily redeployed, primarily tending to patients within their specialty. Some staff are always on duty, others on call.

Short term planning in the JER is therefore highly complex. Prior attempts to predict patient flow using simulation [6, 3] highlight the difficulty of customizing models for a given department. The JER presents a limiting case, since the flow of patients here involves unpredictable and complex interactions between clinical specialties. We use agent-based simulation [16, 12] as a means to reveal and articulate these complex interactions.

Our goal is to provide explanatory insight into JER roles - doctors, nurses, administrators etc. – as they follow triage and other clinical protocols to assign patients to tracks and schedule procedures. By focusing on the clinical agents themselves, we ground our data collection and model development directly into the JER context. This approach minimizes the risk that the data and model are abstracted from the clinical environment, increasing the tractability of model genesis and the credibility of the simulation offered by the model. Clinical event data were gathered in the JER using a custom-designed research instrument. These were combined with data extracted from the Hospital Information System (HIS) (following Mans et. al. [13]) to develop a multi-phase protocol for real-world, time-specific clinical event data acquisition and logging.

Sec. 2 relates our research approach to prior work; in Sec. 3 we outline our method for eliciting simulation agents; in Sec. 4 we set out our data collection methods; in Sec. 5 we present our pipeline of methods and tools used to process the various data types into event logs; in Sec. 6 we demonstrate use of the pipeline in an emergency medicine setting and assess the validity and cohesion of the event log and the elicited agents. The concluding Sec. 7 summarizes our contributions, and explores limitations and future work.

2 Related Work

In this work, we combine agent-based simulation with process mining to elicit a set of agents that supports simulation. This approach is similar to that of Ito et. al. [8] who used process mining to analyze data generated by a multi-agent business simulator. Their extension of an existing multi-agent business simulator to generate data, conversion into event logs and analysis using process mining methods differs from our approach. We gather domain-specific observational event data (from the JER) which is processed into a format that allows it to be synthesized with digital data to create enriched, hybrid event logs. These hybrid logs are used to identify roles and generate simulation agents.

The idea of agent-based simulation is to compose complex systems from agents with a set of constraints for how to behave. Similarly, recent work in process mining has focused on mining process models that are based on temporal logic constraints [4]. The goal explored in this work is to mine constraints in order to elicit agents to be used in future agent-based simulations.

Agent-based simulation has been used for prediction and workflow analysis in emergency departments, showing that agent-based models can simulate bottlenecks and relationships that comprise the workflow [15, 11]. However, these studies offer simulations that though holistic are high-level.

We do not design an agent-based simulation but rather investigate generating one from data. In order to do this we take advantage of generalized formal frameworks from the multi-agent systems community. Development of formal meta-models and frameworks for implementing agent organizations make it feasible to implement systems of agents where agents have roles and collaborative objectives but are capable of acting autonomously. These frameworks are general purpose and thus usable in a range of domains. However, although these frameworks provide a reusable structure for implementing the agents for any domain, significant effort is required to identify the characteristics of a specific domain. Our goal is to use process mining to elicit the agents from historical data in order to reduce the effort needed to create an agent-based simulation and to ameliorate the risks of analytic bias. Our work is further motivated by prominent results of data driven approaches to simulation and decision support in healthcare in the literature [6, 17]. Work in process mining often involve domain experts in the production of models, see for example [14]. We believe reducing the potential of domain expert bias in the produces model is more suitable in order to obtain our goal though.

3 Mining Agents Vision

Agent-based simulation is effective for analyzing complex systems composed of many actors where the behavior of each actor can be described and implemented as an agent using a small set of rules or protocols [12]. However, for social-technical systems with human actors that might work autonomously and/or in collaboration, this is not a trivial task. The identification and implementation of agents in the complex socio-technical *melieu* of the JER is complicated by the interplay of their autonomy - arising from their medical specialisms - and their collaboration in multi-disciplinary activities such as diagnosis. Our overall approach is to use process mining to identify roles, task delegations and capabilities from clinical event data and then translate these into agent organization constraints [5] in a formal framework. The BPMN diagram in Figure 1 shows our agent elicitation pipeline of activities and data artifacts. The figure is adapted from the L^* life-cycle in the process mining manifesto [2, 1]. The steps in our pipeline are:

1. Identify the data needed;
2. Gather data from a real-world case study;
3. Convert the data to event logs using Disco;
4. Create a Petri net for the event log using ProM;
5. Convert the Petri net to a process model using BPMN;
6. Create a candidate role set using a swimlane module;
7. Create a handover of work graph;
8. Create a similar-task graph;
9. Translate the candidate role set into role definitions, handover of work into task delegations and similar-task graph into role capabilities.

Clustering activities using the swimlane module identifies candidate roles. The handover of work graph shows the order of resources performing activities.

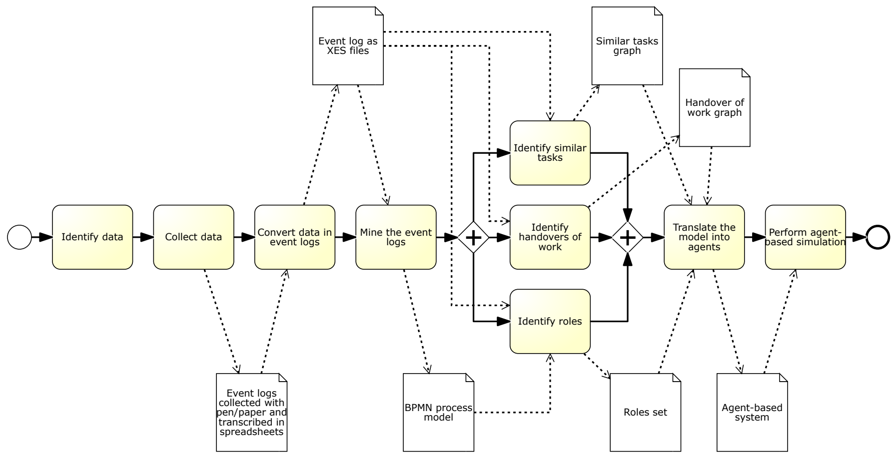

Fig. 1. Agent elicitation pipeline from a real world emergency medicine setting to an agent-based simulation. In order to elicit simulation agents, we use process mining methods and agent-based models for organizations to create a simulation model based on event data. In order to apply process mining, we need an event log that lists a number of instances of the process we want to simulate, and the events with time stamps that took place in each instance. The pipeline also needs to know what staff members are involved in each event and what their position is. Collecting the necessary data requires an understanding of the domain and the scope of available data.

Finally the similar-task graph identifies resources that perform many of the same activities. In the remaining sections we focus on the first two steps but also show how steps 1-8 build cumulatively and discuss their contributions to step 9.

4 Data Collection Method

In this section we present our method for identifying and collecting the data needed to construct the event logs for the agent elicitation pipeline. In order to identify the data, we need to understand the setting from which we gather data and to what extent data is available. For that reason it is critical to involve a clinical expert. We rely on the general guidelines for collecting data for process mining, adapting them for the agent elicitation pipeline.

Long term, we aim to use agent-based simulation for continuously predicting what potential bottlenecks that can occur and help nurses and doctors with preventing the bottlenecks from actually occurring. For this stage of our work we need to identify the contributors to processes enacted in the JER to scope or bound the simulation. Next, we establish which event types we want information about, in particular time stamps and involved agents. Summing up we:

1. Identify which agents to include in the simulation.

2. Identify which activities to include in the simulation.
3. Identify which sorts of events to collection information about.

Next we identify to what extent data is available and how we can collect it. The JER uses a Hospital Information System (HIS) to log data about patient episodes. With guidance from clinical experts, the HIS offers the potential to provide most of the data we need. Alternatively, if data is not available for extraction, we can manually collect data on location. For this we need help from the clinical experts to assess the extent to which we can collect data on location. Collaboratively designing a paper research instrument helped scope and define our data parameters, showing the activities that we are interested in with fields for entering information about staff members and locations, gaining consensus as we worked with the clinical expert.

Having identified the data we need, we decide on how to collect it. Given that we can not extract the data we need from the HIS, we use the form we designed. We discuss with the clinical expert how to fill out the form so that the data collection does not disrupt the workflow of the organization. We also need to establish how much data is necessary to collect. Generally we want as much data as possible but we also need it in a good quality, and if registrations are not done automatically we require help from clinical experts to correct for errors. Doing the data collection over a longer period of time with some time in between each collection allows us to make corrections and improvements, which may not be apparent at first. The overall aim of this aspect of our research design is to achieve a cohesive and comprehensive log for each specific episode, rather than trying to achieve a census of all (undiscriminated) events in the JER. This adds both meaning and validity to the empirical data collection at the case level.

5 Event Log Construction

Having collected the data we need, we convert it into event logs and process models that we can use for creating agent organization constraints. In this section we describe how we processed the data.

To clarify what an event log is, it is a file containing a list of enumerated process instances, i.e. *cases*, where each case contains a series of events with start and end times, plus additional information for each case. An event log is the input for process mining tools such as ProM, and so processing the data to suit the tools we are going to use is critical in order to get meaningful results. In our pipeline we show that the event log is used for 1) generating roles by using a swimlane analysis module, 2) generating task delegations between roles by creating a handover of work graph, and 3) generating capabilities by creating a similar-task graph.

A major point to address in the processing is that existing process mining methods assume that only one resource is involved in each event [1]. In JER however multiple staff members are often involved simultaneously in an event, for example when doctors discuss the course of action for a trauma patient after they have been stabilized. Rather than extending process mining methods to support multiple resources for one event, our solution at this stage of the evolution of our work is to assign names to such groupings. This has the disadvantage that we may lose information about exactly which agents are involved in an event. For the purpose of creating agent organization constraints, we use a naming scheme that combines the positions of the staff members involved.

6 Evaluation

The primary contribution of this phase of our work is the hybrid event log created using the methods outlined above. In this section we evaluate to what extent this type of event log can be used to construct simulation agents for making predictions about patient flow. We do this in two parts. First we discuss the process of applying the method: collecting data and creating an event log, and the content of the log itself. Next we show and discuss preliminary results of applying the three process mining methods for generating roles, task delegations and capabilities.

6.1 Application in a Real-world Setting

Our first step is to identify what parts of the JER that we aim to simulate eventually. In doing so we determine who the agents are and what activities to include in the simulation. The national plan for Danish healthcare services [7] identifies the JER as the common entry point for all acute patients at a hospital. Acute patients are received and treated in JER until they are cleared to either go home or are moved to another department in the hospital. Our scope for agents and activities is limited to those involved from when a patient enters the JER until they leave. We want to collect information about clinical events, which include medical events involving a patient, such as taking samples, and administrative events such as a nurse conveying information to a doctor. We want to include administrative events since they can tell us how tasks are handed over between roles. Next we identify the extent to which data is available and how we can collect it. Here we use the work of Mans et. al. [13] as a starting point, as they describe what kind of data is available in most modern HIS and what kind of data is typically required for different forms of analysis in process mining. We identified the data we needed and met with the hospital management in order to clarify to what extent the data was available in their systems. To that end we designed a paper form that showed what activities, kind of events and information we were interested in. The event form contained fields for the following:

1. Arrival time.
2. Triage time, location and outcome.
3. Additional clinical events.
4. What resources were involved in events, both human and material.
5. What decisions were made and who made them.

6. Information about the finish from JER i.e. when the treatment finished in JER and what the outcome was.

We also wanted data about the staff members who were involved. We designed a separate form to collect this information, so we could simply refer to their ID in the event forms and thus simplify the collection process. The form contained a pseudonymized date field and the following fields for a staff member:

1. Their ID.
2. Their shift.
3. The department were assigned to in their shift.
4. Their position.
5. Their clinical specialties.

With help from the hospital management we found that some, but not all, of the data we needed were present in the HIS. We determined that the system contained the following information of interest to us:

1. Arrival events.
2. Triage events.
3. Treatments, measurements and sampling events.
4. CT, XRAY, blood sample etc. reservations.
5. Times for results from samples and scans.
6. Bed assignments.
7. Movement events within the JER.
8. Request events for additional specialists.
9. What employees were involved in each event.
10. Assignments and reassignments of nurses and doctors.
11. JER departure events.

While the system contained registrations about which nurses and doctors were assigned to patients over time, the potential communication events that could have lead to these registrations were not. We would not be able to tell if or when nurses and doctors communicated in some way beforehand. Such events are of interest in determining the handover of work. In addition, due to data policies, we could not extract data dumps from the HIS. Thus we agreed on having a researcher collect data on location with assistance from a nurse. The researcher could ensure the uniformity of the collected data and the nurse could assist with interpreting the data and identify potentially incorrect or misleading data.

We scheduled data collection in the JER over 9 workdays in order to fully capture the work distribution: patients in the JER are assigned (via triage) to one of three 'tracks', so we planned 3 days in each track: for each of the three tracks, our agenda was as follows

– Day 1: Record data from patients one by one until they leave, following the staff members to register communication events between employees but without interfering with the clinical work.
– Day 2: Copying data from the HIS by hand.

- Day 3: A customized method for collecting data about known parts that have not been covered well during day 1 and day 2.

The reason for using this method is that we could not know beforehand what patients would arrive so having two days with a fixed plan and a third day with an open plan gave us some flexibility.

While collecting the data we observed factors that could be relevant when trying to identify bottlenecks in the JER. We observed 4 different factors: 1) The JER is sometimes used for training nurse students as part of their education, and depending on their level of education more or less time needs to be allocated for the training, 2) some patients have complex treatment paths which require multiple specialties, which involves a doctor handing over the patient to another specialty, 3) preparation and cleanup of beds is done manually and, depending on the circumstances, requires a nurse or cleanup specialist, 4) trauma patients require a team of nurses and doctors from the moment they enter the JER. Having collected the data using paper forms, the next step is to create an event log. As we noted in the description of our method, most events involve multiple staff members. Since the process mining methods assume that one event involve only one resource, we combined the names of the positions of the involved staff members. For example a team of nurses become the resource `NURSES`.

With that approach we made a spreadsheet manually with the following columns and filled it with information in the paper forms:

process id One number for each paper which corresponds to one case.
event Description of the event.
start Start time.
end End time.
team Team name.
place Room ID.
extra Additional information such as what form of communication was used in communication events, what the color code of a triage was, or what treatment was given.
day The day the data was collected. This is a reference to the staff member paper forms, which we used to create the team names.

In doing so we also adjusted some of the labels from what was stated in the paper forms since we changed our labelling scheme over the period we collected the data. This ensures that the quality of the data in the spreadsheet is the same for all days. For start and end times, we only use the time of day, not the date.

The spreadsheet contains 55 complete cases and 597 events. Having the spreadsheet it was straightforward to convert it to an event log using Disco and ProM, annotating the columns with meta-data. The result is an event log with real-world data about staff activity. The log describes in detail the events that happen for a broad variety of episodes in the JER. The event log also does not contain any personal information but contains detailed information for each event about what roles have been involved in the individual events, which is the organizational information we need to generate agent organization constraints for the JER.

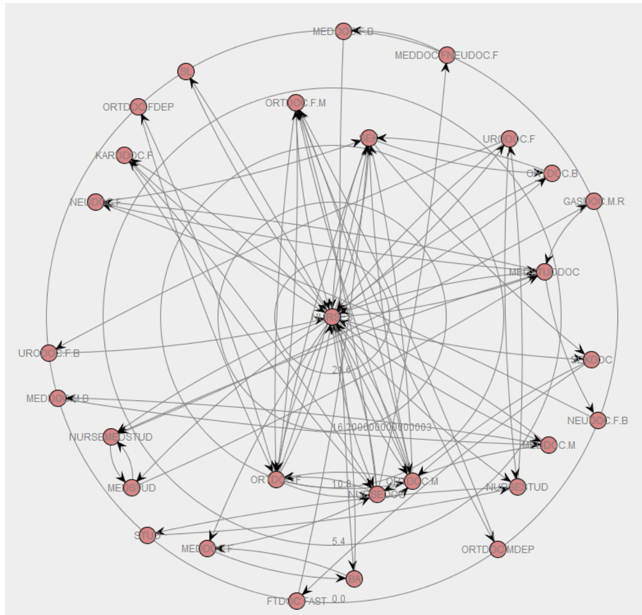

Fig. 2. Handover of work graph. Nodes represent teams and there is an edge between teams if they involved in two sequential events.

6.2 Creating Agent Organization Constraints

Our vision is to use event logs for creating agent organization constraints that can be used for agent-based simulation. In this paper we have focused on creating a real-world event log for this purpose. Although the amount of collected data is low, we would like to discuss preliminary work on the remaining parts of the pipeline showing the potential for eliciting agent organization constraints.

Task Delegation We can generate a handover of work graph as shown in Figure 2 directly from the event log using ProM. The graph shows how the teams take turns being used as a resource for an event in a case. Given that nurses are involved in all cases, it is no surprise that there many edges connect to the `NURSES` node and the centrality of the nurses suggests that addressing nurse shortage would ameliorate bottlenecks. The labelling makes it possible to identify the individual roles involved, for example a nurse is included in both `NURSES` and `NURSEDOC`. The tasks are not shown very clearly in the graph though. Our idea for creating delegation constraints is to translate the graph into logical constraints, which requires identifying the tasks for which the handovers occur.

Capabilities We can generate a similar-task graph as shown in Figure 3 directly from the event log using ProM. The graph shows which teams work on the same

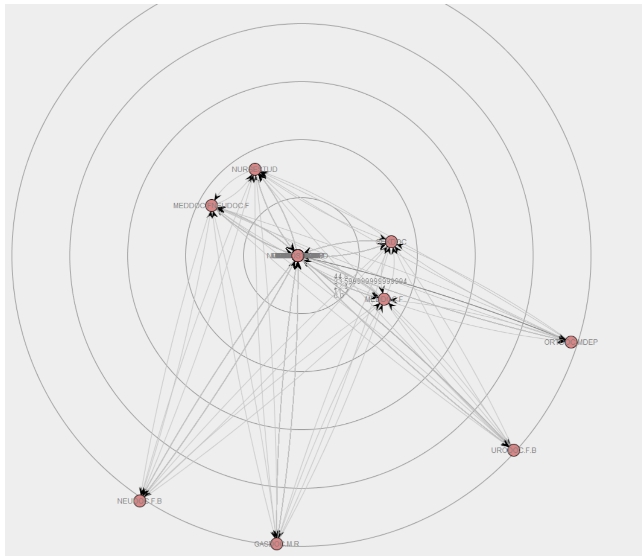

Fig. 3. Similar-task graph. Nodes represent teams and there is an edge between teams if they involved in the same events.

tasks to a high degree. We note that many of the nodes are connected to each other, which is not surprising since we have a small number of event descriptions that are fairly generic for the domain, for example `info` and `treatment`. Creating competence constraints enables us to delineate sets of roles and map the sets to the tasks they can carry out, which requires identifying the tasks that make the roles similar.

Roles We can generate a role-specific business process model (BPM) as shown in Figure 4 by generating a Petri net, BPM and a role set from the event log in ProM. The BPM shows the flow of control in the mined process, and is annotated with swimlanes which splits the BPM into role sets. In our case we only identified one role set, which contains the parts concerning ordering of CT scans. Our idea for creating role definitions is to use the generated role sets as a base. Thus our model originates from unadulterated event data, and our pipeline is agnostic to the *de jure* organizational structures. Our role definitions emerge from the set of objectives that agents enacting that role should achieve.

Simulating an Agent Organization In order to get the best performance, agent-based simulation is often coded from scratch in a general purpose programming language. There are however also agent-based simulation platforms that offer generalized frameworks for coding agents in a high level language which removes the need to handle synchronization issues at a low level. Our vision is to implement the organizational constraints we identify using process mining in

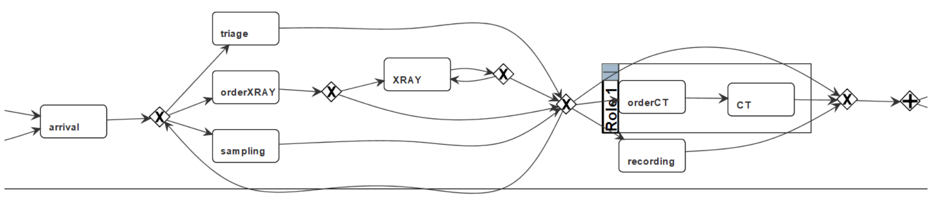

Fig. 4. Role specific process model

such generalized frameworks. To achieve this vision we consider extending work on implementing organizational frameworks in agent-based simulation platforms like GAMA [9] or multi-agent system platforms like Jason [10].

7 Conclusion and Future Work

We have presented our vision for eliciting simulation agents from real-world event logs by using process mining methods. Figure 1 illustrates our vision as a pipeline of activities and emergent products. In this paper we have focused on the creation a hybrid event log and how it can be used as the basis to develop agent-based prediction models which are agnostic to organizational structures. We created a log that gives a snapshot of reality which is not statistically significant but still offers meaningful insight into reality. We gathered and analyzed data using the protocols summarized in Figure 1 to offer a proof-of-concept for our approach. These preliminary data also offered meaningful insights into the location of handovers that have the potential to ameliorate bottlenecks in the clinical setting.

For each event in the event log we have assigned a team resource that identifies what roles are involved in an event. To do this we preprocess the data, combining the positions of staff members into a team name so that we can recover the individual roles later when creating agent organization constraints. Alternatively, if the process mining tools supported multiple resources for one event, we could omit this part of the preprocessing and potentially also simplify the translation into agent organization constraints. Future work includes investigating such options.

Time constraints limited the scope of our observations, analysis and modeling. Future work will extend the scale of our models, offering further validation of the methods set out in this paper as the pipeline is more fully loaded. This will enable further exploration of translation of process mining output to agent organization constraints, the preliminary steps for which have been set out in this paper. We also plan to implement these constraints in agent-based simulation, building further on the philosophy of structural agnosticism by using generalized agent organization frameworks, and thus generate an agent-based prediction model from data.

Acknowledgements This work is supported by the PhD project between PDC A/S and the Technical University of Denmark (DTU). We thank the institute Region H, which manages the hospitals in the Danish capital region, for providing input for the use case. Christopher Davis is Otto Mønsted Visiting Professor at DTU Compute, supported by the Otto Mønsteds Fond. Supported by the Process-Oriented Data Science for Healthcare Alliance (PODS4H Alliance).

References

1. van der Aalst, W.: Process Mining: Data Science in Action. Springer, 2 edn. (2016)
2. van der Aalst, W., et al.: Process mining manifesto. In: Business Process Management Workshops. pp. 169–194. Springer (2012)
3. Almagooshi, S.: Simulation modelling in healthcare: Challenges and trends. Procedia Manufacturing **3**, 301–307 (2015)
4. Burattin, A., Maggi, F.M., Sperduti, A.: Conformance checking based on multi-perspective declarative process models. Expert. Syst. Appl. **65**, 194–211 (2016)
5. Dignum, V., Padget, J.: Multiagent organizations. In: Weiss, G. (ed.) Multiagent Systems, pp. 51–98. MIT Press (2013)
6. Goienetxea Uriarte, A., Ruiz Zúñiga, E., Urenda Moris, M., Ng, A.H.: How can decision makers be supported in the improvement of an emergency department? a simulation, optimization and data mining approach. Operations Research for Health Care **15**, 102–122 (2017)
7. Hansen-Nord, M., Steensen, J.P., Holm, S.: Process driven patient tracks in FAM. Scand. J. Trauma. Resusc. Emerg. Med. **18(Suppl. 1)**, 27 (2010)
8. Ito, S., Vymětal, D., Šperka, R., Halaška, M.: Process mining of a multi-agent business simulator. Comput. Math. Organ. Theory **24**(4), 1–32 (2018)
9. Larsen, J.B.: Adding organizational reasoning to agent-based simulations in GAMA (2018), revised and extended with 13 pages to appear in EMAS 2018 post-proceedings (Springer)
10. Larsen, J.B.: Going beyond BDI for agent-based simulation. Journal of Information and Telecommunication (TJIT) pp. 1–19 (2019). https://doi.org/10.1080/24751839.2019.1620024
11. Liu, Z., Rexachs, D., Epelde, F., Luque, E.: An agent-based model for quantitatively analyzing and predicting the complex behavior of emergency departments. Journal of Computational Science **21**, 11–23 (2017)
12. Macal, C.M., North, M.J., Samuelson, D.A.: Agent-based simulation. In: Gass, S.I., Fu, M.C. (eds.) Encyclopedia of Operations Research and Management Science, pp. 8–16. Springer, Boston, MA (2013)
13. Mans, R.S., van der Aalst, W.M.P., Vanwersch, R.J.B.: Process Mining in Healthcare: Evaluating and Exploiting Operational Healthcare Processes. Springer (2015)
14. Rovani, M., Maggi, F.M., De Leoni, M., Van Der Aalst, W.M.: Declarative process mining in healthcare. Expert Systems With Applications **42**(23), 9236–9251 (2015)
15. Wang, L.: An Agent-based Simulation for Workflow in Emergency Department. IEEE Systems and Information Engineering Design Symposium pp. 19–23 (2009)
16. Wooldridge, M.: Intelligent agents. In: Weiss, G. (ed.) Multiagent Systems, pp. 3–50. MIT Press (2013)
17. Zhou, Z., Wang, Y., Li, L.: Process mining based modeling and analysis of workflows in clinical care. Proceedings of the 11th IEEE ICNSC pp. 590–595 (2014)