



# Integrated Declarative Process and Decision Discovery of the Emergency Care Process

Steven Mertens<sup>1</sup> · Frederik Gailly<sup>1</sup> · Diederik Van Sassenbroeck<sup>2</sup> · Geert Poels<sup>1</sup>

Accepted: 11 October 2020 / Published online: 24 October 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Deviations and variations are the norm rather than the exception in medical diagnosis and treatment processes. Physicians must leverage their knowledge and experience to choose an appropriate variation for each patient. However, this knowledge and experience is often tacit. Process modeling offers a way to convert tacit to explicit knowledge. Many process mining techniques have been developed due to the difficulty of doing this manually, yet, they often neglect the decisions themselves, and these proposed techniques are just one piece of a comprehensive process discovery method. In this paper, we use the Action Design Research methodology to develop a method for process and decision discovery of medical diagnosis and treatment processes. The method was iteratively improved and validated by applying it to a practical setting, which was the emergency medicine department of a hospital. An analysis of the resulting model shows that previously tacit knowledge was successfully made explicit.

**Keywords** Process discovery · Process mining · Decision mining · Knowledge extraction · Healthcare · Knowledge-intensive

## 1 Introduction

Within the Information Systems domain it is generally accepted that a process-oriented approach centered around process modeling results in efficiency gains, efficacy gains and/or cost reduction (Dijkman et al. 2016). Typical examples are applications in sectors such as manufacturing (Hertz et al. 2001), sales (Kim and Suh 2011) and software development (Krishnan et al. 1999). However, the healthcare sector is one the exceptions (Lenz and Reichert 2007; Palvia et al. 2012). This is surprising because some of the main concerns trending

in eHealth are very similar to those of these other sectors, namely, cost reduction, efficiency and patient orientation (Payton et al. 2011). Healthcare processes can be subdivided in two groups of processes: medical diagnosis/treatment processes and organizational/administrative processes (Lenz and Reichert 2007). The slow adoption of process modeling in healthcare is primarily manifested for the medical diagnosis and treatment processes. These processes typically represent the extreme end of the complexity spectrum for processes, hypercomplexity (Klein and Young 2015), and can be characterized as dynamic, multi-disciplinary, loosely framed, human-centric and knowledge-intensive processes (Mertens et al. 2017; Rebuge and Ferreira 2012). Loosely framed processes have an average to large set of possible activities that can be executed in many different sequences (Di Ciccio et al. 2015a), leading to situations where deviations and variations are the norm rather than the exception (Mardini and Ras 2020). The knowledge workers participating in the execution of a process (i.e., physicians and other healthcare personnel) use cognitive processes to decide which activities to perform and when they are to be performed (Greenes et al. 2018). To make these decisions, they often leverage what is called tacit knowledge, meaning that they have an implicit idea of the appropriate actions to perform when certain conditions apply (Lenz and Reichert 2007). This idea might be partially based on explicit knowledge (e.g., medical or hospital guidelines) but experience usually factors in heavily. This kind of tacit

---

✉ Steven Mertens  
steven.mertens@ugent.be

Frederik Gailly  
frederik.gailly@ugent.be

Diederik Van Sassenbroeck  
diederik.vansassenbroeck@azmmsj.be

Geert Poels  
geert.poels@ugent.be

<sup>1</sup> Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium

<sup>2</sup> Department of Emergency Medicine, AZ Maria Middelaers, Buitenring Sint-Denijs 30, 9000 Ghent, Belgium

knowledge is also collective in the sense that it often cannot be traced back to just one individual, but rather is (unevenly) spread across organizational units of knowledge workers that share the same or similar experiences (Kimble et al. 2016).

*“To make medical knowledge broadly available, medical experts need to externalize their tacit knowledge. Thus, improving healthcare processes has a lot to do with stimulating and managing the knowledge conversion processes.” (Lenz & Reichert, 2007, p. 44)*

Modeling these processes can be beneficial in many ways. A process model can serve as a means to document the process (which increases transparency for all stakeholders), to communicate changes explicitly, educate students and new process actors, to offer passive process support (i.e., using the model as a roadmap during execution) and as a foundation for active process support and monitoring (e.g., in process engines) and all sorts of management and analysis techniques (e.g., bottlenecks, process improvement...). Process modeling in healthcare has traditionally focused on clinical pathways (CPs) (a.k.a. clinical guidelines, critical pathways, integrated care pathways or care maps) (De Bleser et al. 2006; Campbell et al. 1998; Combi et al. 2015; Heß et al. 2015; Huang et al. 2014; Kamsu-Foguem et al. 2014; Lawal et al. 2016; Rotter et al. 2010; Zhang et al. 2015). Although the exact definition of what they entail can vary from paper to paper (Vanhaecht et al. 2006), the general consensus seems to be that they are best practice schedules of medical and nursing procedures for the treatment of patients with a specific pathology (e.g., breast cancer). CPs can be considered process models, but they have serious drawbacks. CPs are the result of Evidence Based Medicine, which averages global evidence gathered from exogenous populations that may not always be relevant to local circumstances (Hay et al. 2008). Therefore, CPs can be great general guidelines, but real-life situations are typically more complex. Second, comorbidities and practical limitations (e.g., resource availability) can prevent CPs from being implemented as-is. Third, the real process also comprises more than just the treatment of patients, as steps like arriving at a diagnosis and other practicalities should be considered as well. A CP is therefore a form of high-level model of one idealized process variation. To unlock the full aforementioned benefits, a more complete view on the real processes is needed that encapsulates all possible care episodes within a certain real-life management scope (e.g., department). Of course, CPs and other medical guidelines can subsequently be used to verify whether such model of the real process follows the medical best practices as closely as possible.

Modeling healthcare processes manually, for example by way of interviewing the participants, can be a tedious task due to their intrinsic complexity and the extensive medical knowledge involved. This often leads to a model that paints an

idealized picture of the process that overly simplifies reality (Antunes et al. 2020). Meanwhile, hospital information systems already record a lot of data about these processes for the sake of documentation (Bygstad et al. 2020). The sheer amount of data in these systems makes manual analysis unrealistic. Consequently, there has been an increased interest into process mining techniques that can automate the discovery of healthcare process models from such data sources (Abo-Hamad 2017; Andersen and Broberg 2017; Baker et al. 2017; Basole et al. 2015; Duma and Aringhieri 2020; Funkner et al. 2017; Helm and Paster 2015; Huang et al. 2016; Kovalchuk et al. 2018; Orellana Garcia et al. 2015; Rojas et al. 2019; Wang et al. 2017) and recent literature reviews indicate that it is still trending upwards (Ghasemi and Amyot 2016; Kurniati et al. 2016; Rojas et al. 2015, 2016). However, the existing work is almost exclusively focused on techniques that result in either imperative process models or CPs. The introduction of new process modeling languages based on the declarative paradigm (Hildebrandt et al. 2012; Pesic 2008) opens up new research avenues (Cognini et al. 2018). Declarative languages are better suited to represent loosely framed processes because they can model the rules describing the process variations instead of having to exhaustively enumerate all variations as imperative techniques do. The knowledge-intensive side of the processes can be supported by integrating a knowledge perspective into the declarative process modeling languages (Burattin et al. 2016; Kluza and Nalepa 2019; Mertens et al. 2017; Santoro et al. 2019; Schönig et al. 2016). This allows for the decision logic of the process to be modeled and makes the resulting group of languages a natural fit for loosely framed knowledge-intensive healthcare processes. Declarative process mining techniques have been applied before on healthcare processes, but not on the complete medical care process (e.g., only the treatment of already diagnosed breast cancer patients) (Maggi et al. 2013) and without considering the decision logic that governs the process (Burattin et al. 2015; Rovani et al. 2015).

From a methodological perspective, methods for the application of data mining have been developed and are widely used (Mariscal et al. 2010; Shearer 2000). However, these are too high-level and provide little guidance for process mining specific activities (van der Aalst 2011). Therefore, methods were developed specifically for the process mining subdomain (van Eck et al. 2015; Rebuge and Ferreira 2012; Rojas et al. 2019). These provide a good foundation for typical process mining applications but are geared towards imperative process mining techniques and ignore data and decision aspects of the process. The method of Rebuge and Ferreira (2012) is positioned to target healthcare processes, but the use of imperative process mining techniques means that they have to resort to clustering techniques in an effort to filter out some behavior and isolate smaller parts of the process. (Rebuge and

Ferreira 2012) also represent the method as a straightforward linear approach, which is unrealistic in the context of real-life applications on loosely framed processes.

This paper reports on an Action Design Research cycle that is part of a Design Science project to develop a process-oriented system that manages and supports loosely framed knowledge-intensive processes (Mertens et al. 2017, 2018; Mertens, Gailly, and Poels 2019a; Mertens, Gailly, Van Sassenbroeck, et al. 2019b). The research artifact presented in this paper is a method for process and decision discovery of loosely framed knowledge-intensive processes. This method consists of the preparation of process data to serve as input for a decision-aware declarative process mining tool called DeciClareMiner (Mertens et al. 2018), the actual use of this tool and the feedback loops that are necessary to obtain a satisfactory model of a given process. The method was iteratively improved and validated by applying it to a real-life case of an emergency care process.

The remainder of this paper is structured as follows. Section 2 presents our research methodology. This is followed by a brief overview of the modeling language DeciClare and the corresponding process mining tool DeciClareMiner in section 3. Section 4 describes the method for process and decision discovery of loosely framed knowledge-intensive processes. The method application is summarized in section 5. And finally, section 6 concludes the paper and presents directions for future research.

## 2 Research Methodology

Action Design Research (ADR) (Sein et al. 2011) is a specific type of Design Science Research (Hevner et al. 2004). While typical Design Science Research focusses solely on the design of research artifacts that explicitly provide theoretical contributions to the academic knowledge base, ADR simultaneously tries to solve a practical problem (Sein et al. 2011). The practical problem serves as a proving ground for the proposed artifact and a source for immediate feedback related to its application. Based on this feedback, the design artifact is iteratively improved. In this paper, the research artifact and practical problem are, respectively, the method for process and decision discovery of loosely framed knowledge-intensive processes and the creation of a process and decision model of a real-life emergency care process.

An ADR-cycle consists of four stages:

- Stage 1: Problem formulation
- Stage 2: Building, intervention and evaluation
- Stage 3: Reflection and learning
- Stage 4: Formalization of learning

The previous section discussed the problem statement: the lack of a process mining method for declarative process and decision mining techniques. Therefore, the research artifact of this paper is a method for process and decision discovery of loosely framed knowledge-intensive processes. This method should consist of the preparation of process data to serve as input for a decision-aware declarative process mining tool, the actual use of this tool and the feedback loops that are necessary to obtain a satisfactory model of a given process.

For the second stage, we have performed an IT-dominant building, intervention and evaluation stage, which is suited for innovative technological design. This stage consists of building of the IT-artifact (i.e., the envisioned method), an intervention in a target organization (i.e., an emergency medicine department), and an evaluation. We started from the existing methods (van Eck et al. 2015; Rebuge and Ferreira 2012) to identify relevant steps in a process mining project. These activities were subsequently evaluated for their applicability to a declarative setting. As the artifact is still in preliminary development, we have limited the repeated intervention steps to light-weight interventions in a limited organizational context (Sein et al. 2011).

The research artifact was evaluated after each execution of the second stage. The evaluation was performed by the researchers in close cooperation with the head of the emergency medicine department of the hospital. It consisted of two parts, each of which highlighted a different use case of the artifact. The head of the emergency medicine department took on the role of the domain expert, who could identify problems with the application of the research artifact which, in turn, were traced back to the research artifact itself by the researchers. A discussion of the results of the evaluation followed. Potential changes to the research artifact in the context of the practical application were proposed in the first half of these discussions, which corresponds to the reflection and learning stage of the ADR-cycle (i.e., stage 3). The proposed changes were then discussed in the more general context of process and decision mining projects for loosely framed knowledge-intensive processes in the second half of these discussions, which corresponds to the formalization of learning stage of the ADR-cycle (i.e., stage 4). Finally, the resulting changes were implemented to end each iteration of the ADR-cycle. A new iteration was initiated if significant changes were made to the artifact during the previous iteration.

## 3 Background

DeciClare (Mertens et al. 2017) is the language we used in the project for process modeling. While many currently available process modeling languages focus solely on the explicit sequencing of the activities of processes (i.e., the control flow of the process activities), DeciClare offers integrated functional,

control-flow and data perspectives. This makes it possible to model the decision logic that governs the sequencing of activities (e.g., activity Y must be executed because activity X was executed and data attribute ‘d’ has value 19). The decision logic is typically tacit knowledge in many loosely framed knowledge-intensive processes (Wyatt 2001), which heavily influences the information needs of the knowledge workers as well as the process execution itself (Stefanelli 2004). By modeling or mining this decision logic, the tacit knowledge is transformed into explicit knowledge that can be used, managed, analyzed and reused. In addition, DeciClare is fully declarative modeling language that also includes hierarchical activity modeling and a resource perspective.

Previously in this project, we developed a tool that is able to (semi-)automatically and efficiently mine declarative process and decision logic models of loosely framed, human-centric and knowledge-intensive processes: DeciClareMiner (Mertens et al. 2018). The underlying algorithm consists of two phases. In the first phase it mines decision-independent constraints (Table 1 for explanations of the used process mining/modeling terminology) using a search strategy inspired by the Apriori association rule mining algorithm (Agrawal and Srikant 1994). The search strategy is complete, which means that all decision-independent constraints that fit the given parameters are guaranteed to be found. The second phase uses a heuristic belonging to the class of genetic algorithms to link seed constraints with activation decisions in order to find decision-dependent constraints. Support for the automatic discovery of a resource perspective is still under development for DeciClareMiner (nor is it available in any of the other declarative process mining tools), so this paper will focus mostly on the other perspectives.

## 4 A Method for Process and Decision Discovery of Loosely Framed Knowledge-Intensive Processes

We first created an initial method by combining elements from the existing process mining methods (van Eck et al. 2015; Rebuge and Ferreira 2012) and adapting them to a declarative process mining context (Fig. 1). Note that we did not include a process analysis, improvement or support step (as in PM<sup>2</sup> of van Eck et al.). This is because we see these more as separate usage scenarios of the results of process discovery (i.e., the created event log and process model). Such usage scenarios could be part of more general process mining method(ologie)s, but our focus was solely on a method for process discovery (Dumas et al. 2018). The internal structure or language (grey text) of the inputs and outputs (gold) as well as the entity responsible for the execution (blue text) of each step (green) have also been added. Optional activities are represented by transparent and dashed rectangles. During different

iterations of the ADR-cycle, several missing intermediate activities and iteration loops were identified that were needed to discover a satisfactory process and decision model. The result is presented in Fig. 2. The remainder of this section describes each separate step of the method and the intermediate activities they entail.

### 4.1 Extraction

The information system(s) (IS) of an organization contain(s) the data needed to execute all processes in the organization. Typically, it contains data related to many different processes. The Information Technology (IT) department, that is responsible for the IS, must first identify the data related to the target process in close collaboration with the consultant and extract it from the IS database(s). Of course, that is in the assumption that the required data is being stored. Table 2 summarizes the ideal data and the minimal data requirements to enable high-quality results.

If the minimum requirements from Table 2 cannot be met, there will be significant limitations to the results of the method application. This will often be the case in real-life applications with data sources that are process-unaware (i.e., systems that are oblivious to the process context in which data processing takes place) (Dumas et al. 2018). Typical issues include missing events and inaccurate timestamps (e.g., the stored timestamp corresponds to when the data is entered in the system but deviates from when the activity actually took place). However, meaningful results can still be achieved when these limitations are properly considered or mitigated. An increased adoption of more process-aware information systems would certainly be beneficial for process mining projects, as these can provide the required data directly or will at least do a better job at storing the required data (i.e., the ideal data from Table 2).

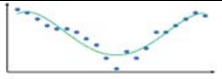
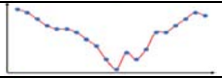

### 4.2 Data Preprocessing

The use of a certain type of database management system (DMBS) (e.g., a relational DBMS) imposes specific structural limitations on how data is stored in the database (e.g., data attributes must be single-valued in a relational DBMS). These limitations should not hinder the subsequent steps of the proposed method, and therefore, a conceptual model should to be constructed that makes abstraction of any structural limitations of the original database and other implementation-related aspects (Fig. 3).

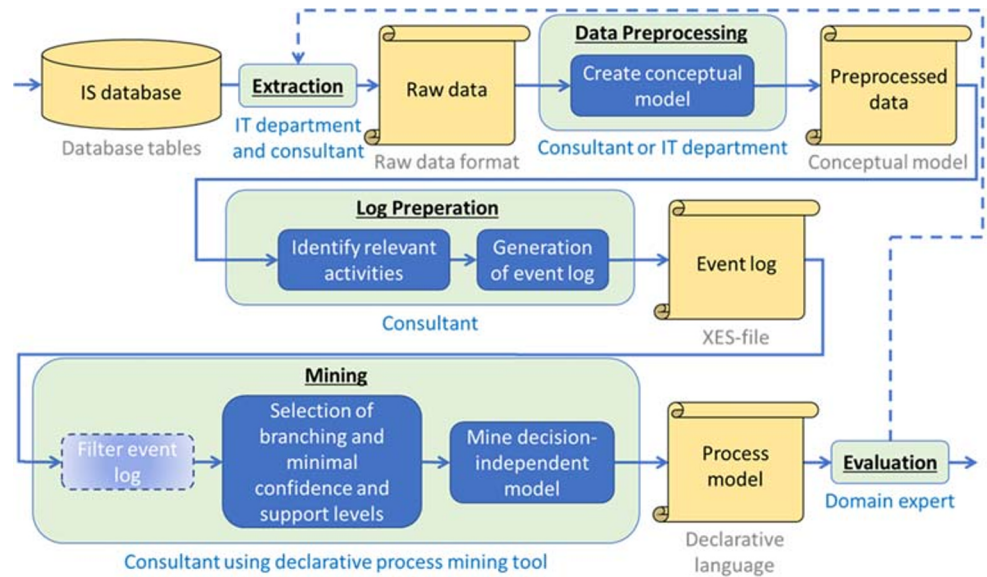
Aside from database limitations, a typical difficulty of a method such as this one is that much of the data is stored as plain text in the database. This data cannot be used without introducing some additional structure. Text mining and Natural Language Processing (Manning and Schütze 1999) techniques can be used to transform the plain text to more



**Table 1** Explanations of key process mining and/or declarative modeling terminology

<i>Event</i>	An occurrence of something that happened during the execution of a process. We make a distinction between activity and data events. The former represents the start, end or failure of the execution of an activity by a process actor. The latter represents the moment when a certain data value was made available to the process actors. Resource events are outside of the scope of this project.	
<i>Trace</i>	Each individual execution of the process as logged by the information system(s) of an organization. It consists of a sequence of activity events performed and the data events that were generated during the execution of the process instance. In healthcare, a trace aligns with what is called an episode of care for one patient.	
<i>Event log</i>	A collection of many traces relating to the execution of one specific process.	
<i>Declarative constraint</i>	A limitation of the allowed behavior when executing a process. This can be both in a positive manner (i.e., a certain activity or event is required to be executed at least a given number of times during a certain time span) and negative manner (i.e., a certain activity or event is prohibited to be executed during a certain time span). The types of constraints can be classified according to the perspective to be modeled: functional/control-flow, data and resource constraints. In this paper we will treat existence (e.g., aforementioned positive and negative examples) and relational constraints (i.e., Response(A, B) states that if A is executed, then B must be executed somewhere in the future before the process execution ends) as reflecting the functional/control-flow perspective.	
<i>Data attribute</i>	The definition of some data property in the form of a name and a data type (e.g., the Boolean ‘hasPaid?’).	
<i>Data value</i>	A specific value that a certain data attribute can have (e.g., TRUE).	
<i>(De)activation decision</i>	A set of data values connected with logical operators (i.e., AND/OR) that represent a specific process context. For example, certain behavior might apply when the patient is diagnosed with hypothermia or when the patient is alcohol intoxicated and cold (written as [[hypothermia], [alcohol intoxication, cold]] in DeciClare).	
<i>Decision-independent constraint</i>	A constraint that has a trivial (i.e., always true) activation decision. This means that there are no data conditions that need to apply for the constraint to be active. Therefore, this kind of constraints must always be satisfied by a process execution. For example, ‘Patient registration’ must be executed at least once. Note that this term only refers to the data conditions of the activation decision. A relational constraint can have both an activation decision based on data conditions and a specific activity or event (e.g., A in Response(A, B)) that must have occurred before the target becomes required (e.g., B in Response(A, B)).	
<i>Decision-dependent constraint</i>	A constraint that does not have to be satisfied until the activation decision evaluates as true. It needs to be satisfied starting from when it was activated up until it (optionally) gets deactivated when the deactivation decision evaluates as true. An inactive constraint can be violated with no consequences. For example, the constraint ‘after an examination by a physician, surgery has to be performed at least once’ and an activation decision that evaluates as true when it is a trauma case and when the patient exhibits signs of shock and/or heavy local inflammation or pain during the examination.	
<i>Conforming activation</i>	A constraint that is activated and satisfied by a certain (partial) trace (e.g., consequence side of an activated relation constraint is satisfied).	
<i>Violating activation</i>	A constraint that is activated and violated by a certain (partial) trace (e.g., consequence side of an activated relation constraint is not satisfied).	
<i>Conformance percentage (a.k.a. confidence) of a constraint</i>	The number of traces with a conforming activation for the constraint divided by the total number of traces in which the constraint was activated.	
<i>Support percentage of a constraint</i>	The percentage of traces in which the constraint was activated (independent of whether it is a conforming or violating activation).	
<i>Branching level of a constraint</i>	The number of activity/event parameters of most DeciClare constraints can be increased from level 1 (e.g., at least one CT scan must be performed) to higher levels by introducing logical conjunctions and disjunctions (e.g., level 2: at least one echo or CT scan must be performed).	
<i>Seed constraint</i>	A decision-independent constraint that has more than enough support but insufficient confidence. For example, Response(A, B) states that if A is executed, then B must be executed at least once somewhere after A. Let us assume minimal support and confidence levels of 10% and 90%, respectively, that A occurs in 15% of the traces of the given event log (i.e., a support level of 15%) and that in 80% of those traces A was eventually followed by B (i.e., a confidence level of 80%). The example constraint is a seed constraint. Adding an activation decision to this seed constraint can result in a constraint with a higher confidence level and a (slightly) lower support level, because it only applies to a subset of the original traces.	
<i>Model fit</i>	How well a model fits the set of given observations. This typically directly correlates to its predictive power for future observations.	
<i>Overfit</i>	A model that corresponds too closely or exactly to the data used for its creation. It may therefore fail to fit additional data or predict future observations reliably.	
<i>Underfit</i>	A model that corresponds too little to the data used for its creation. It may therefore fail to encapsulate the logic of the underlying phenomenon with enough details to predict future observations reliably.	

**Fig. 1** The initial method for process discovery based on existing methods



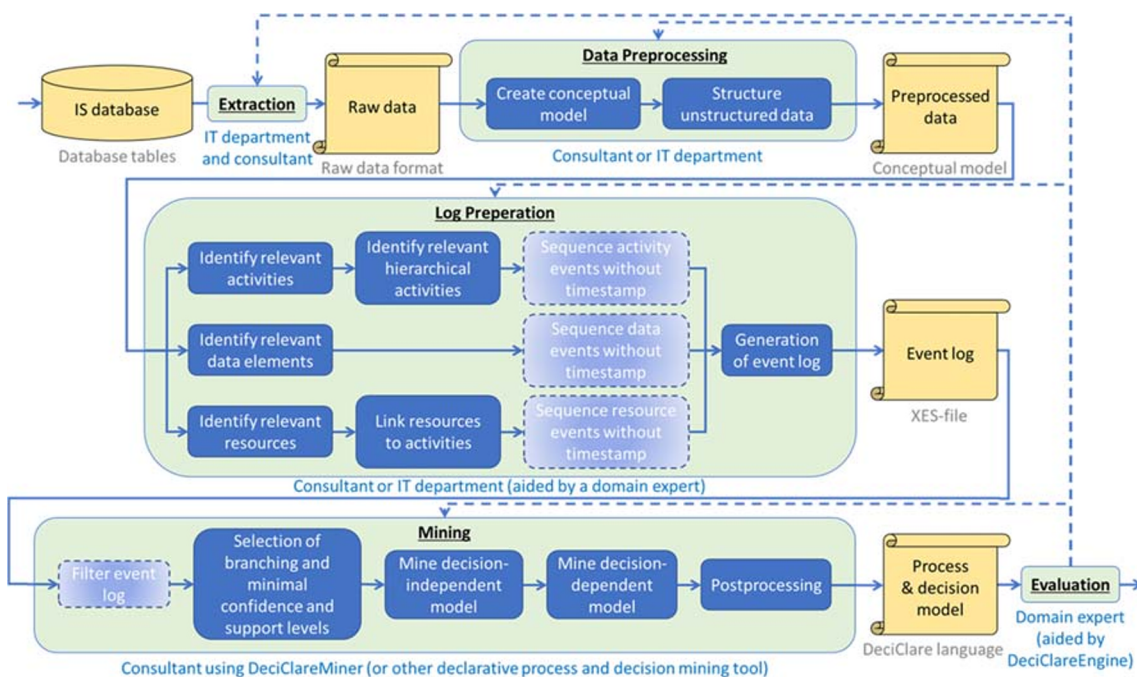
structured data, for example TiMBL (Daelemans et al. 2009). Of course, different techniques can be applied without changing the proposed method.

### 4.3 Log Preparation

In this step, the traces of an event log describing the data need to be reconstructed (Fig. 4). This means walking through the data of each historic execution of the process, as structured by the conceptual model, and identifying the relevant activity, data and resource events.

The hierarchical extension of DeciClare allows activities to be grouped as higher-level activities, which in turn can be grouped as even higher-level activities. For example, the activities of ‘Request an echo’, ‘Request a CT’ and ‘Request an MRI’ can be grouped in an activity ‘Request a scan’, which in turn can be grouped with ‘Request lab test’ as the activity ‘Request diagnostic test’. An activity can also be part of multiple higher-level activities. The definition of higher-level activities will often require some help from a domain expert to ensure domain validity.

We also included three optional intermediate activities in Fig. 4 (denoted transparently and with dashed lines). These are



**Fig. 2** The final method for process and decision discovery for loosely framed knowledge-intensive processes

**Table 2** The ideal data and the minimal data requirements to get high-quality results

	Ideal	Minimum requirements
Activity events	(Pre)defined events that relate to the activities executed in the process.	Enough secondary data to reconstruct the events that relate to the activities executed in the process (i.e., reverse engineering).
Data events	Data related to the decision rationale of, and the source data used by, the process actors when determining the process variant to follow. This includes all data related to the specific context and the defining characteristics of each case.	All documented data that was used by the process actors when determining the process variant to follow. So, all available data related to the specific context and the defining characteristics of each case. A process and decision model can be discovered if relevant data is available, but otherwise it can only be a process model.
Resource events	(Pre)defined events that relate to the availability, usage and authorization of resources during executed in the process.	Enough secondary data to reconstruct the events that relate to the availability, usage and authorization of resources during executed in the process (i.e., reverse engineering).
Timestamps	Timestamps for each activity event and data event, which represent the exact time that the corresponding real-life activity was performed or that the corresponding data element (e.g., lab results) became available to the process actors.	A relative order of the activity and data events that represents the order in which the corresponding real-life activities were performed and the corresponding data elements (e.g., lab results) became available to the process actors.
Number of cases	Representative set of cases during a period in which no seriously disruptive changes to the process occurred.	

not needed when the minimal requirements from Table 2 are met. However as noted in subsection 4.1, this is not always possible in real-life projects. Often there will be some missing events or timestamps in the dataset. These optional intermediate activities facilitate the resolution or mitigation of such data quality issues as much as possible.

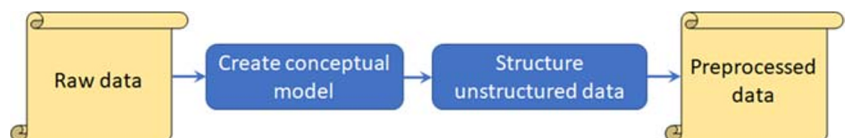
#### 4.4 Mining a Process and Decision Model

Process mining tools can use the event log composed in the previous steps as input for the discovery of process models (Fig. 5). Process mining techniques that create imperative process models often resort to the use of slice and dice, variance-based filtering (e.g., clustering) or compliance-based filtering to reduce the complexity of a given event log (van Eck et al. 2015; Rebuge and Ferreira 2012; Rojas et al. 2019). For process mining techniques that create declarative process models this is not a necessity, as these can handle more of the complexity, but can also optionally be used to focus on a specific part of the dataset.

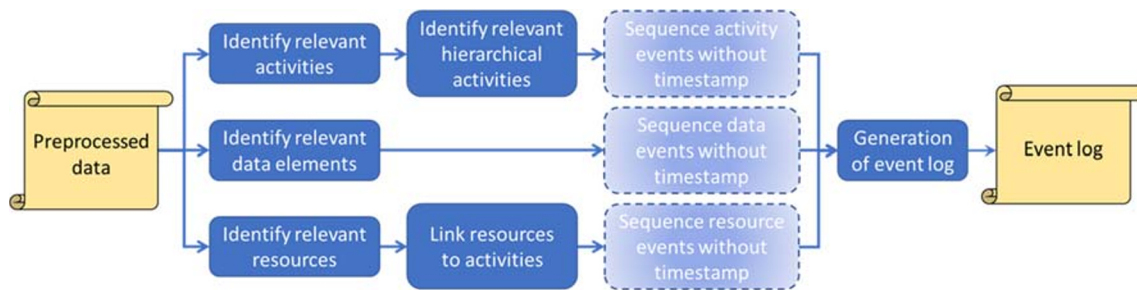
While the other activities can be performed mostly automatically, the selection of the maximal branching level and minimal confidence and support levels (see Table 1 for definitions) is more of a judgement call. The minimal confidence and support levels are directly linked to the data quality. These parameters can be used to control the level of detail and

complexity of the resulting models in order to prevent overfitting and manage the impact of noise in the event log. Therefore, the consultant responsible for the project should choose good initial values for these parameters. Generally, we would advise to start with a confidence level of 100% (or very close to it) and a minimal support level that corresponds to tens or even hundreds of traces (i.e., 0.1%–2% for a log of 10.000 traces). This will result in a model describing behavior that is never violated in any trace and is reasonably frequently occurring. These levels can be tuned in later iterations based on the results of the evaluation described in the next subsection. If the resulting model contains too many decision-dependent constraints and/or overfitted decision logic, then a step-by-step increase of the minimal support level could improve the results. On the other hand, decreasing the minimal support level is a better course of action when insufficient decision logic was found. The minimal confidence level can also be gradually lowered to counteract the noise in the dataset (i.e., real behavior is not found due to data quality issues or just extreme outliers). A minimal confidence level lower than 100% will allow the mined constraints to be violated by some of the traces in the event log if enough other traces confirm the corresponding behavior pattern. Although DeciClareMiner can mine constraints with lower-than-100% confidence, it can result in inconsistent models (Di Ciccio et al. 2015b) and tools to resolve this (Di Ciccio et al. 2017)

**Fig. 3** The intermediate activities to go from the raw data to preprocessed data







**Fig. 4** The intermediate activities to go from the preprocessed data to an event log

are not yet available for DeciClare. This means that it can be useful to mine constraints lower-than-100% confidence when the goal is to explain some specific behavior or to expose some pieces of knowledge for human understanding. However, if the goal is to use the mined model as input for a business process engine of a process-aware information system, then constraints lower-than-100% confidence should only be considered after extensive verification. The maximal branching level should preferably be chosen after consulting domain experts and/or process actors. A branching level of one is often the best starting value. Whenever possible, hierarchical activities should be used as they are more computationally efficient and could eliminate the need for higher branching levels.

The second phase of DeciClareMiner, the search for decision-dependent rules, can be executed multiple times with the same parameter settings, as this phase is not deterministic. The miner contains tools to combine these separate results to one big model at the end. If no data events were defined in the event log, the mining of a decision-dependent model can also be skipped. This will result in a (declarative) process model without any data-related decision logic.

Postprocessing is applied automatically to the raw output of DeciClareMiner. This eliminates equivalent decision rules (i.e., combinations of data values that uniquely identify the same cases) and returns a much smaller model with the given minimal support and confidence levels. However, it is not necessarily true that when two decision rules are equivalent in a given event log, that this is also true in the corresponding real-life process. It can be impossible to distinguish between a real-life decision rule and other decision rules that identify the same set of cases due to the incompleteness of the log (i.e., not every possible case occurring). For example, consider the constraint that at least one NMR will be taken. This could apply to patients with a red color code and a remark that mentions

the neck, but the same patients might also be identified uniquely with a red color code and the remark that mentions sweatiness (just because this limited set of patients happened to have both).

#### 4.5 Evaluation

After each mining step, the correctness of the reconstructed event log and the mined DeciClare-model needs to be evaluated by a domain expert (see Fig. 2). Verifying the correctness of the reconstructed event log is the first priority, as it is the input for the mining techniques. Random sampling can be used to select a set of reconstructed traces that will be manually reviewed by domain experts, preferably side-by-side to the original data so that the experts get the full context. This should be repeated until all following stopping criteria are reached:

- 1 There should be no discrepancies between the original timeline and the reconstructed sequence of activities of the simulated patients (i.e., in the correct order or at least in a realistic order when the original data does not specify an exact time).
- 2 The available contextual data should be contained by the reconstructed data events. Of course, The focus here is on the essential contextual data that is relevant to the decision-making of process actors.
- 3 All data events of the simulated patients should be based on data that was actually available at that time in the process (i.e., no data events using data that was added in retrospect or at an unknown time).

When the reconstructed event log is deemed satisfactory, the evaluation of the mined models can begin. The mined process and decision models are difficult to review directly because there is often nothing tangible to compare them with



**Fig. 5** The intermediate activities to go from the event log to an integrated process and decision model



(i.e., tacit knowledge). The understandability issues associated with declarative process models (Haisjackl et al. 2014) and the typical size of these models also make it infeasible to go over them one-by-one. The type of evaluation needed also depends on the specific use case that the organization has in mind for the mined model.

In general, we would advise the adoption of a similar approach to (Braga et al. 2010; Guizzardi et al. 2013), which uses the visual simulation capabilities of Alloy Analyzer to evaluate a conceptual model, for the simultaneous evaluation of a reconstructed event log and the corresponding mined model. The DeciClareEngine (Mertens, Gailly, and Poels 2019a) (Fig. 6) and its ‘Log Replay’-module (Fig. 7) can replace Alloy Analyzer to demonstrate what the mined process model knows about what can, must and cannot happen during the execution of specific traces. It can show what activities have been executed at each stage during the replay of a trace (top), the corresponding data events (middle) and the activities that can be executed next according to the mined process model (bottom). The user can view the reasons why a certain activity must/cannot be executed as well as any

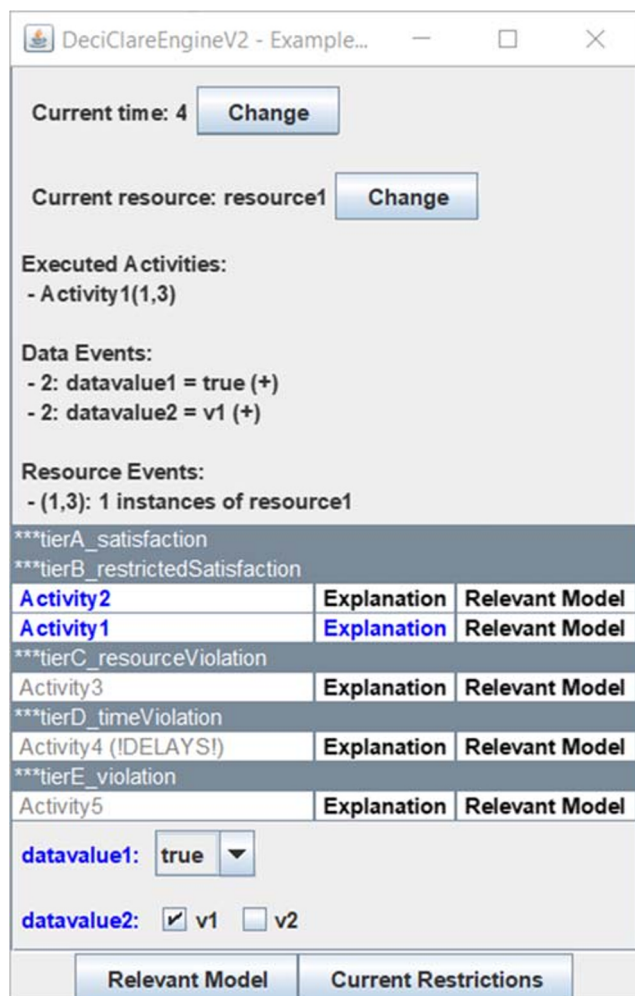


Fig. 6 The interface of the DeciClareEngine model simulator

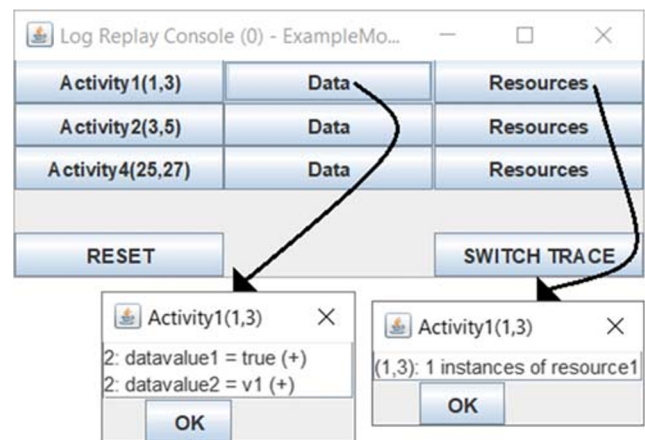


Fig. 7 The ‘Log Replay’-module for DeciClareEngine

future restrictions that might apply (in the form of constraints) at any time by clicking on the ‘Explanation’-, ‘Relevant Model’- and ‘Current Restrictions’-buttons. Domain experts can be asked to give feedback during the replay of a random sample of reconstructed traces. Before the execution of each activity, the questions from Table 3 can be asked. The answers gauge the correctness of the reconstructed event log (i.e., 1, 2 and 3) and mined model (i.e., 4 and 5) as well as the value of the model to the managers and physicians (i.e., 6).

Ideally, the model fit should be perfect. However, this is not realistic for loosely framed knowledge-intensive processes. The main obstacles are the data quality and completeness of the event log. On the one hand, data of insufficient quality can hide acceptable behavior and introduce wrong behavior in models (Suriadi et al. 2017). On the other hand, event logs are almost always incomplete (i.e., they do not contain examples of all acceptable behavior) (Rehse et al. 2018). Thus, some (infrequent) behavior is bound to be missing in the mined models, even when inductive reasoning is applied to generalize the observed behavior. The result is that mined models often will simultaneously contain some over- and underfitting pieces. While a slight underfit is not that big of a problem for process automation, overfitting can be. Overfitting causes acceptable behavior to be rejected because it was not observed in exactly the same way in the reconstructed event log. We envision that mature process engines will automatically relax or prune away these overfitting rules (i.e., self-adaption) when a user overrides the model, similar to how it would handle gradual process evaluation (Deokar and El-Gayar 2011). This also reduces the need for a full validation of the mined model as it will be validated gradually by the users, with each overfitting rule serving as a warning that the deviating behavior is not typical. Of course, this is not a free pass to purposefully overfit models. It just means that when searching the optimal model fit, a small overfit is manageable and perhaps preferred to an underfit. This feature is currently in development for DeciClareEngine.

**Table 3** The questions for the domain experts and the underlying purpose behind each of them for the evaluation

Question	Purpose
1 Based on the data, are there any activities that might be missing at this point?	<i>Identify missing activities</i>
2 Is the order of activities how you would expect based on the data?	<i>Identify activity reconstruction mistakes</i>
3 Do the data events generated so far correspond to the data available at this point?	<i>Identify data reconstruction mistakes</i>
4 Does the model allow activities to be executed next that should not be allowed?	<i>Detecting underfit in model</i>
5 Does the model prohibit activities to be executed next that should be allowed?	<i>Detecting overfit in model</i>
6 Does the decision logic for why certain activities are allowed/prohibited reflect real decision-making processes?	<i>Measuring knowledge extraction of model</i>

## 5 Method Application: The Emergency Care Process

This section provides a brief description of the application of the method presented in the previous section. The practical problem and organizational context that has been tackled is the discovery of an integrated process and decision model of the emergency care process that takes place in the emergency medicine department of a private hospital with 542 beds.<sup>1</sup> Data about the registration, diagnosis and treatment of patients over a span of just under 2 years is available in their Electronic Health Record (EHR) system. Most of this data is plain text (in Dutch), which is a typical obstacle for process and decision discovery (Sittig et al. 2008). The role of consultant was performed by the first author as is typical in ADR.

Discovery methods are used as a tool to achieve specific goals in an organization. This method application contained two of the most important use cases for model discovery of loosely framed knowledge-intensive processes in general: a model for operational process automation and a model to extract tacit knowledge related to operational decision-making.

For the first use case, the discovered model should enable operational process automation by serving as the foundation for a context- and process-aware information system (Nunes et al. 2016). Such a system can support the process actors by enforcing the model (Mertens, Gailly, and Poels 2019a) and by providing context-aware recommendations on what to do next (Mertens, Gailly, Van Sassenbroeck, et al. 2019b). While the former can prevent medical errors, the latter can offer guidance and help optimize the process. The primary goal of the method application was to create a model that facilitates the transition from a process-unaware information system to a context- and process aware information system without disrupting the current working habits. A review of whether every patient in the dataset was handled correctly according to the current medical guidelines was outside the scope of the project, therefore, we assumed that all patients in the dataset were handled correctly. This meant that the goal for the

resulting model was to encapsulate the observed behavior from the reconstructed event log and also allow unobserved behavior that follows similar logic, while at the same time restricting other unobserved behavior as much as possible.

Medical personnel typically leverage their knowledge and experience when deciding on how to deal with a specific patient. This knowledge can apply to all patients (e.g., registration always first) or to a specific subset of patients based on the characteristics of the patient and other context variables (e.g., if shoulder pain and sweaty, then take echo of the heart). This kind of knowledge is typically not explicitly available anywhere, but rather contained in the minds of the physicians. The second use case consists of transforming such tacit knowledge into explicit knowledge, which can be used in many ways: it can be discussed amongst physicians to promote transparency and ensure uniformity, used to train new physicians and nurses, to improve resource planning (e.g., can reserve some slots for patients that will need an MRI based on their contexts, even though it has not been formally requested yet), etc. Therefore, the goal for the resulting model was to discover useful and realistic medical knowledge.

### 5.1 Extraction

The hospital provided us with an export of six database tables from their internally developed EHR system spanning the period from 31/12/2015 until 22/09/2017. The data was exported from the original relational DBMS in the form of anonymized CSV-files. The minimum requirements from Table 22 were not satisfied because of some missing timestamps and a lack of data related to the availability, usage and authorization of the resources during the execution of the process.

### 5.2 Data Preprocessing

Figure 8 presents the conceptual model that was reconstructed from the raw EHR data. A care episode represents a single admittance of a patient that can end with a discharge or a transfer to another department. The plain text data fields were structured by transforming them each to a set of keywords.

<sup>1</sup> AZ Maria Middelaes hospital, Ghent, Belgium - <https://www.mariamidelaes.be/>

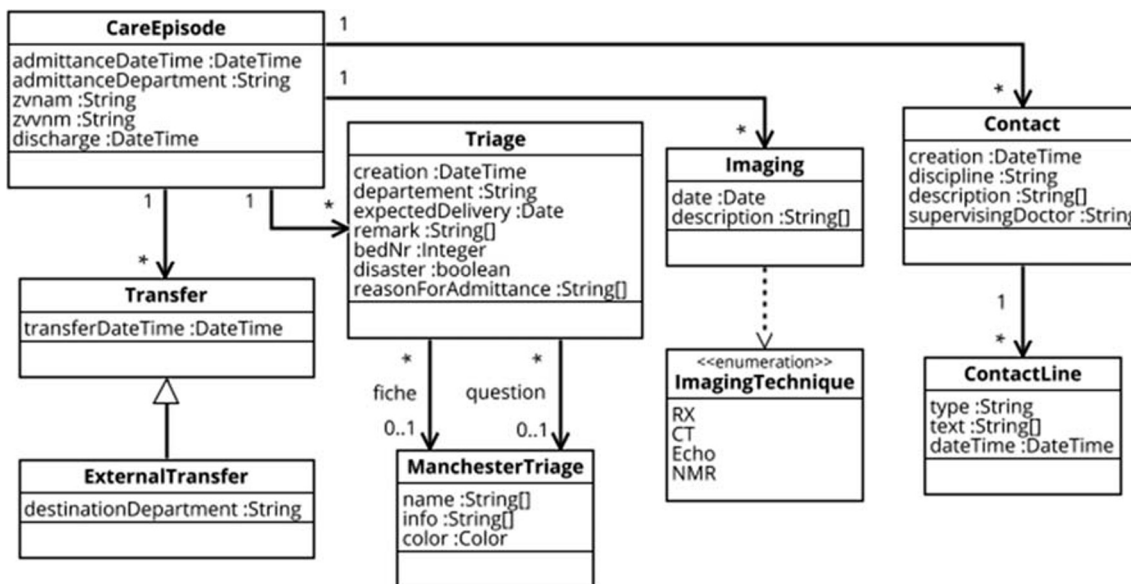


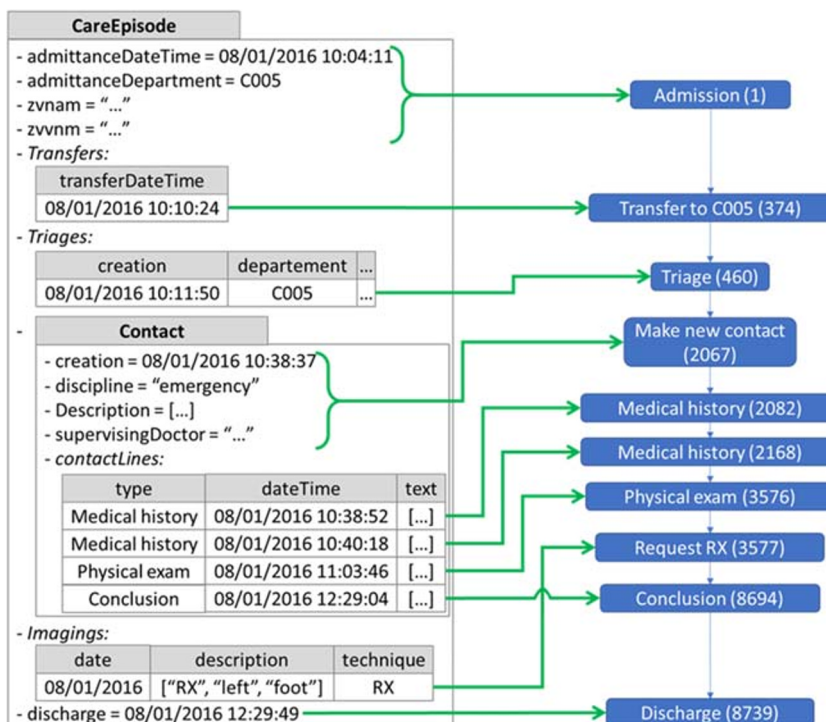
Fig. 8 UML class diagram of the conceptual model for the preprocessed data

### 5.3 Log Preparation

Next, we reconstructed the chronology and created a trace for each care episode. The reconstruction of the activity events is illustrated with an example in Fig. 9. The activity events are shown in blue with the elapsed time between brackets (corresponding data events are omitted). The green arrows link the activity events with the data on which they are based. The relevant data values are transformed to data events in the trace. The most interesting

data to reconstruct the decision logic of the process is stored in the plain text attributes (e.g., the attribute ‘text’ in the ContactLine-class), which have been preprocessed to multisets of keywords as described in subsection 4.2. For each of those attributes, the keywords were transformed to categorical data events (e.g., ‘Description of Medical history’ = ‘foot’). Practically, we used the XES-format for the event log and used a custom extension to include general data events (i.e., adds data events with a timestamp that do not need to be directly linked to an activity event).

Fig. 9 An example of how the internal representation (preprocessed data) is transformed to a trace





Several iterations were needed to create a satisfactory event log. Yet, the need for these iterations only can come up during later steps because a domain expert must make this assessment. It would be possible to add an intermediate evaluation of the constructed event log at the end of this step, depending on the urgency of the project and the availability of the domain expert. However, the next step (mining) is automated anyway. Therefore, we decided against adding the extra evaluation step. We just used the created event log to mine a model and had the domain expert evaluate the event log and mined model simultaneously during the normal evaluation step. The first iterations of the event log creation focused on the identification and granularity of activities and data elements, while the sequencing of activities with missing timestamps (i.e., imaging activities and lab tests) also required multiple iterations. Each iteration incrementally improved the event log to better reflect reality. The resulting event log contains 41,657 traces using 116 unique activities and 951 unique data values. Some additional statistics about the log are presented in Table 4.

#### 5.4 Mining a Process and Decision Model

DeciClareMiner was executed on an Intel Xeon Gold 6140 with the emergency care event log from subsection 5.3 and several different parameter settings as input. The maximal branching level was set at 1 (i.e., no branching). Some smaller scale experiments with higher branching levels showed a significant increase in model size with little added value. The definition of 15 hierarchical activities also reduced the need for higher branching levels. We started with a minimal confidence level of 100% (i.e., only never-violated behavior was mined) and a minimal support level of 10% (i.e., only behavior that was observed in minimally 4166 of 41,657 traces). These two values were iteratively adjusted to improve the quality of the results. Minimal support levels were relaxed from 10% to 5%, 2%, 1%, 0.24%, 0.048%, 0.024%, 0.012% and 0.007%. These values were not chosen randomly, but rather based on the minimal number of observations we wanted for the discovered behavior patterns (e.g., 0.007% corresponds to a minimum of three observations). In the context of the second use case, we also mined models with minimal confidence levels of 98%, 97.5%, 96.67%, 95% and 90% in order to also find knowledge patterns that are otherwise hidden due to noise in the data. The results<sup>2</sup> consist of a post-processed model for the evaluation of the first use case and the raw model used for the evaluation of the second use case.

#### 5.5 Evaluation

We used DeciClareEngine and its ‘Log Replay’-module for the evaluation of the model of the emergency care process. We

also created an additional module to visualize the original data plain text of the EHR side-by-side with the trace and its keyword representations of those texts to facilitate the evaluation of the reconstructed event log. The ‘Log Replay’-module and the new custom module have an interface that gives an overview of the activities of the trace on the left and ‘Data’-buttons on the right (left side of Fig. 10). Clicking on one of the activities will set the state of the engine to just after the execution of that activity, while the original EHR data can be opened by clicking on the ‘Data’-button next to each activity (right side of Fig. 10). This was used to replay the care episodes of many patients for the domain expert. He was asked the questions from Table 3 before and after each activity in a replayed trace. Hence, it was an evaluation using sampling and validation of the model in the context of specific patients, and not on a per constraint basis. The evaluation helped identify several issues with the reconstructed event log as well as with the mined models. These issues were handled during subsequent iterations of the method. This was repeated until a satisfactory result was reached (taking into account the data quality of the available dataset).

The result of the first use case, the discovery of a model for operational process automation, is an operational model of the emergency care process that takes place in the emergency medicine department of the AZ Maria Middelaers hospital in Belgium. The model contains 28,162 decision-independent and 6283 decision-dependent constraints. The minimal confidence level setting of 100% and the application of induction by the mining technique resulted in a model that was sufficiently flexible and general to serve as input for a business process engine (e.g., DeciClareEngine) of a context- and process-aware information system. Figure 11 presents a simplification of the decision-independent constraints of the model using the Declare visual syntax (Pesic 2008). It shows only the activities done by emergency physicians (so no specialist activities), combines the external transfers that do not have a specific relation constraint into a generalized activity called ‘External transfer (other)’, omits chain constraints and omits other constraints with a support lower than 0.25%.

The model created for the first use case provides a rare peek into the daily operation of an emergency medicine department. The model reveals the patterns and the specific contexts that some of the patterns apply to. The broad outlines of the model were not surprising to any of the process actors, as they already had an implicit idea of the general way things are done at the department, but the model makes them explicit so that it becomes accessible to outsiders (e.g., management, emergency physicians in training, patients, etc.). However, from the point of view of knowledge extraction things get more interesting when diving deeper into a model. The model from the first use case is limited to behavior that is not violated in any of the observations in the event log in order to be suited for operational use. This means that even the slightest inaccuracy

<sup>2</sup> <https://github.com/stevmert/discoveryMethod>



**Table 4** Additional statistics on the event log

	Total of all traces	Minimum per trace	Maximum per trace	Average per trace	Median per trace
Duration (hours)	133,686	0.067	22.3	3.2	2.9
#Activity events	625,682	3	128	15	17
#Data events	1,350,331	2	166	32.4	32

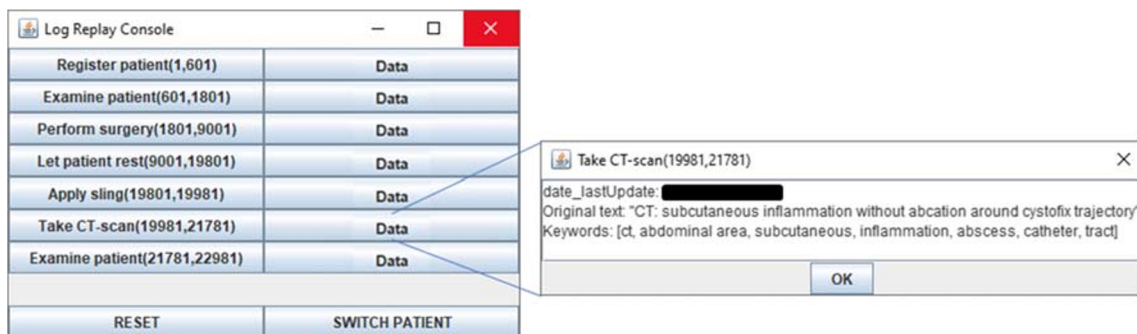
of the source data, due to either forgetfulness or incompetence, will prevent the discovery of certain valid behavior and decision patterns no matter its frequency of occurrence.

The knowledge currently captured by a model mined with DeciClareMiner consists of two main components: the functional and control-flow perspectives describing the occurrence and sequencing rules concerning the activities and the decision logic that determines the context in which each occurrence and sequencing rule applies. The second use case focusses on the knowledge that was extracted from the reconstructed event log about the operational decisions made by the process actors. This refers to the data contexts that make constraints active, which in turn can require or block the execution of certain activities (e.g., if the triage mentions an overdose and respiratory issues, then the patient will eventually be transferred to the intensive care unit). The goal was to ascertain whether or not the discovered decision logic matched the tacit medical knowledge of the physicians. The model did not need to be executable for this use case. Therefore, it combined the results of the first use case with additional mining results with minimal confidence levels lower than 100%.

We used an evaluation consisting of two parts for this second use case. The same setup as for the first use case was used in the first part because this allowed for an efficient validation of the reconstruction of the event log as well as showed some of the activation decisions in a real context. The second part entailed a direct evaluation of the activation decisions of selected constraints. The disjunct pieces of decision logic were listed for each of the constraints, which facilitated a Likert-scale scoring by domain experts based on (medical) correctness of each activating data context. The domain

expert was subsequently asked to evaluate some of the discovered decision criteria for when certain activities need to be executed. For example, if the patient received a green color code during the triage and the medical discipline of the care episode is ophthalmology, then the patient would eventually be discharged. Note that this type of rules (i.e., positive) are the most interesting but also the most difficult to discover. A lot of decision logic related to negative rules was also discovered, but this is less interesting due to those rules being less helpful and typically common sense (e.g., a patient with dementia should never be transferred to Neonatology). The domain expert evaluated 208 discovered decision criteria for when the following activities are executed at least once: Patient Discharge, Request CT, Request RX, Transfer to Pediatrics, Transfer to Oncology/Nephrology, Transfer to Cardiology and Transfer to Geriatrics. Each disjunct part of the discovered decision logic was listed and subsequently scored on a Likert-scale ranging from medical nonsense to realistic decision logic. The results are summarized in Fig. 12. We also included the results of an intermediate version of this evaluation to show the evolution after several iterations of the proposed method.

The results show that around 53% of the decision criteria were evaluated as realistic in the final evaluation, although some small nuances were often missing. This highlights the potential value of the proposed method to automatically transform tacit to explicit knowledge in healthcare, and more generally, for all sorts of loosely framed knowledge-intensive processes. The improvements over the intermediate evaluation demonstrate the need for multiple iterations. The model used in the intermediate evaluation contained much more nonsense,



**Fig. 10** The log replay console that can replay a trace up to every decision point and display the original EHR data for the simulated patient

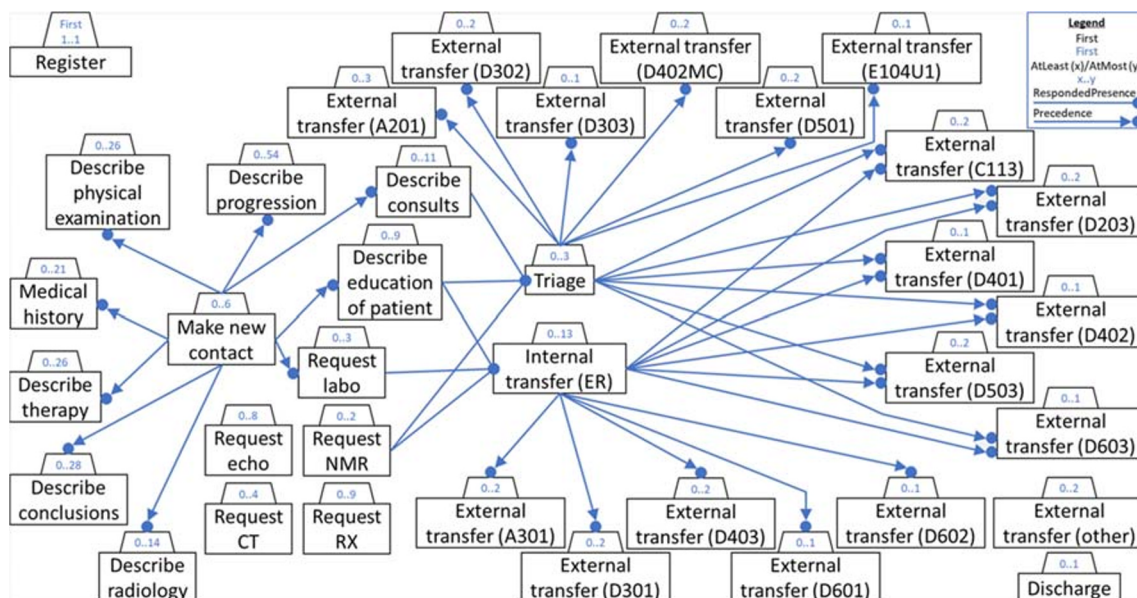


Fig. 11 Simplified Declare representation of the mined process model

primarily due to the use of lower minimal support levels during the mining step. Much of this nonsense was filtered out of the model by increasing the minimal support level.

## 5.6 Discussion

The paper is based on the premise that imperative modeling techniques are not suitable for loosely framed processes (van der Aalst 2013; van der Aalst et al. 2009; Goedertier et al. 2015; Mertens et al. 2017; Rovani et al. 2015), which includes many healthcare processes. Figure 13 presents an imperative process model mined from the same event log created in this section.<sup>3</sup> The model was created by the Disco-tool,<sup>4</sup> which was also used to mine a model for an emergency medicine department in Rojas et al. (2019). It contains all activities but ignores the data events (as this feature is not supported by the tool) and shows just the 1.1% most frequently travelled paths. This is a good example of how traditional process discovery techniques would represent the emergency care process of the AZ Maria Middelaes hospital. Despite the omission of data events and 98.9% of the less frequently travelled paths, the result is still an unreadable model that is often referred to as a spaghetti model (van der Aalst 2011; Rojas et al. 2019). This is caused by the 28,980 unique variations in the log (out of 41,657 patients), as identified by Disco when only considering the activity events, of which the most frequent variation occurs just 686 times (1.65%). An imperatively process discovery tool either interconnects almost every activity to include each of these variations (i.e., unreadable spaghetti model) or filters out all complexity to only leave some sort of

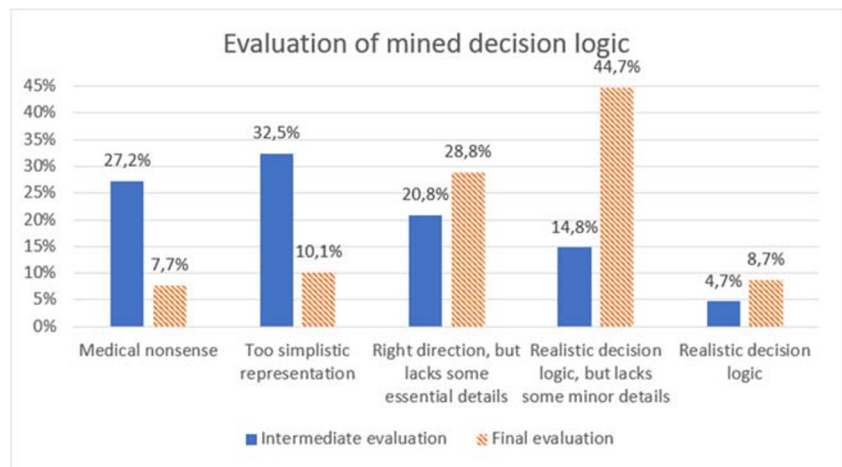
generalized happy path (i.e., does not reflect reality). This is a typical problem when modeling healthcare processes imperatively that results in poor usability of the resulting models (Duma and Aringhieri 2020; Fernandez-Llatas et al. 2015). Rojas et al. (2019) worked around this by asking domain experts to manually define subsets of activities that are expected to be part of smaller and more manageable subprocesses, while Duma and Aringhieri (2020) resort to making significant assumptions and simplifications. Declarative modeling languages are better suited to these processes (Goedertier et al. 2015), because they actually extract the logical rules that define the connections between activities instead of just enumerating the most frequently encountered connections like imperative techniques do. The DeciClare language used in our method caters specifically to loosely framed knowledge-intensive healthcare processes with its integrated data perspective, which enables even more fine-grained knowledge to be extracted from the data.

The process and decision model created during the application of the proposed method certainly revealed some useful insight into the emergency care process of the AZ Maria Middelaes hospital. Despite the basic (rather than advanced) data analysis and other limitations, it managed to transform a lot of tacit knowledge into valuable and explicit knowledge. Of course, much of the discovered knowledge was already known to the experienced physicians in the department, but now it is also readily available to less experienced physicians and other stakeholders that want to gain insight into the operation of an emergency medicine department. Having explicit rules for the occurrence and sequencing of activities and the corresponding decision logic available enables deeper analysis and optimization of the working habits of the emergency physicians. Both use cases showed the potential of the

<sup>3</sup> Also available at <https://github.com/stevmert/discoveryMethod>.

<sup>4</sup> <https://fluxicon.com/disco/>

**Fig. 12** The results of the evaluations of the mined decision logic by the domain expert



proposed method to make previously tacit knowledge explicitly available for all stakeholders, for both operational and decision-making purposes. Process and decision models, like the one mined in this paper, have the potential to unlock a new tier of applications further along the line. For example:

- An explicit justification as to why certain activities are performed in specific cases and others are not.
- The decision logic of treatments in the model can be compared to the available clinical guidelines. Do the physicians follow the guidelines? If not, what are the characteristic and decision criteria of the cases in which they diverge from the guidelines? Are these justifiable?
- The decision logic of different physicians can be compared. Do the physicians treat similar patients similarly? What are the differences? Can adjustments be made to make the service level more uniform?
- The knowledge contained by models will make simulations much more realistic and flexible. For example, how would a department address an increase in a group of pathologies by 5%? Would this create new bottlenecks? Can these be eliminated by changing, if possible, some decision criteria?
- The knowledge made explicit by the mined models can be used to bring interns more quickly up to speed or even to educate future nurses/physicians as part of their medical training.
- The mined models can be used for evidence farming (Hay et al. 2008). This is the practice of posteriori analysis of clinical data to find new insights without setting up case-control studies. Evidence farming can be used as an alternative or, even better, as a supplement to evidence-based medicine.

Throughout the application of the developed method, we encountered several typical limitations of process discovery projects. Different types of noise in the data that was used to create the model are the primary limitation. This is real-life

data that was recorded without this sort of applications in mind, so it would be unrealistic to expect perfect data. The data quality framework of Vanbrabant et al. classifies 14 data quality problems typical to electronic healthcare records of emergency departments in several (sub)categories: missing data, dependency violations, incorrect attribute values and data that is not wrong but not directly usable (Vanbrabant et al. 2019). All 14 data quality problems were encountered at one point or another during this project. In terms of the forms of noise defined by van der Spoel et al., we encountered the following (van der Spoel et al. 2013):

- **Sequence noise:** errors in, or uncertainty about, the order of events in an event trace.  
 In this project we did not have an exact timestamp of the radiology and lab activities. So, a workaround was devised that is by no means perfect. Additionally, the timestamps that we did have are those of when the physicians or nurses describe activities in the EHR, but this does not necessarily reflect the timing and order of the actual activities performed. Some physicians prefer to update the EHR of a patient for several activities simultaneously. Consequently, the preservation of the sequence information is dependent on whether physicians describe those activities in the same order as they performed them (which was generally the case). Although there is no way to fully resolve this issue, aside from a whole new system to record the data, we did use names for the activities that correctly reflect the data that we had. Many activities were named like ‘Describe ...’ or ‘Request ...’ to signify that we cannot make a statement about when the actual activity occurred, just when the description or request was made.
- **Duration noise:** noise arising from missing or wrong timestamps for activities and variable duration between activities.

The issues described for sequence noise also apply here. We only had timestamps of when an update was

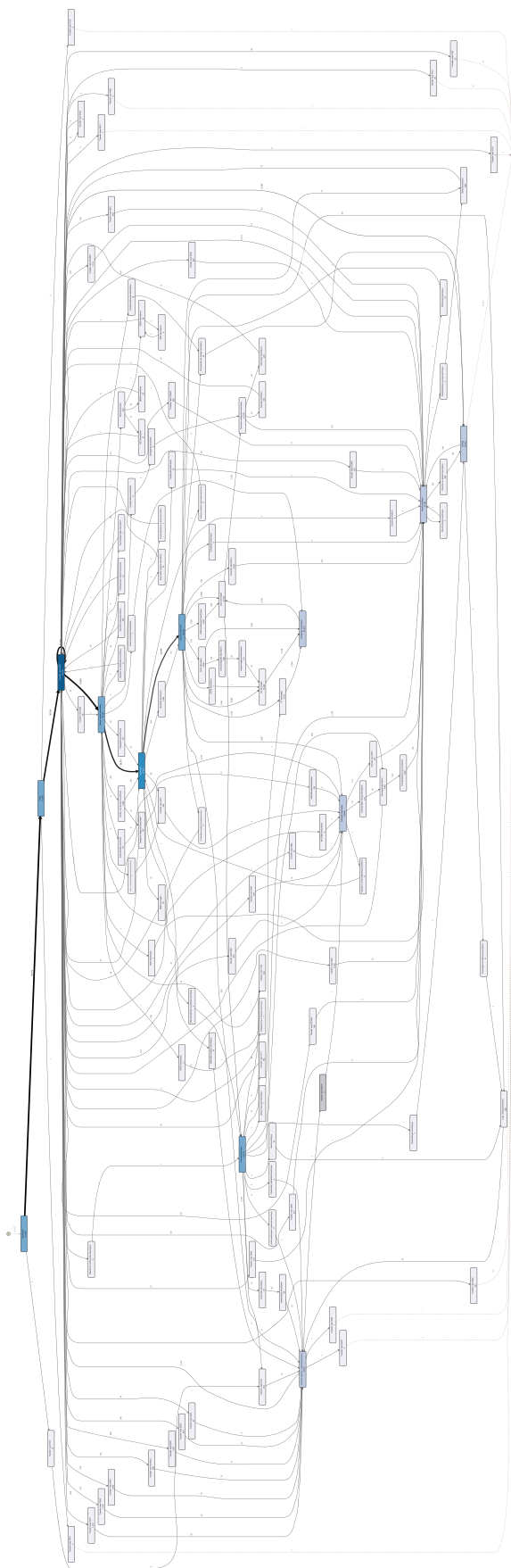


Fig. 13 The imperative process model as generated by the Disco-tool

saved to the EHR. Hence, there was no timestamp of the actual activity or even a duration. We gave every activity an arbitrary duration of 2 s for practical reasons, but this does not reflect reality.

- **Human noise:** noise from human errors such as activities in a care path which were the result of a wrong or faulty diagnosis or from a faulty execution of a treatment or procedure.

When writing in the EHR, physicians and nurses will unavoidably make mistakes: typos, missing text, missing activity descriptions, activity descriptions entered in the incorrectly activity text field... But even correct description of activities can cause problems as a different jargon will make it more difficult to detect patterns. We preprocessed the plain text data to resolve the more frequent typos and used a list of synonyms and generalizations to deal with the use of different jargons. However, these solutions are not exhaustive, and thus, could still be improved. The release of an official dictionary and thesaurus of medical jargon in the Dutch language would be a big step in the right direction.

Two other general limitations related to data completeness typical for process mining projects also applies to this project. Firstly, process mining starts with an assumption of event log completeness (Ghasemi and Amyot 2016). This means that we assume that every possible trace variation (i.e., type of patient, diagnosis, complication, treatment, etc.) that can occur also occurred in the event log. Of course, this is not realistic as each event log contains just a subset of these variations. The result is that the process models will generally overfit the event log (i.e., it was never observed that a patient needed more than four CT-scans, so the model says this cannot occur, while there might be infrequent cases for which this could be possible). This can be mitigated during the project by fiddling with the minimal support level for the mining step and after the project by slowly gathering more and more data. The second general limitation relates to the completeness of the context data. Decision mining works best when all the data used by the process actors during the execution of the process is available in the event log. However, due to privacy concerns this is difficult to achieve, especially in healthcare. In our case, we did not get access to several general patient attributes (e.g., age and gender) as well as to any information about the medical history of the patients. Of course, this severely limits what can be discovered. And finally, we encountered the following event log imperfection patterns (Suriadi et al. 2017): form-based event capture, unanchored events, scattered events, elusive cases and polluted labels.

The issues described above lead to a loss of accuracy in the resulting process and decision model. Some knowledge is lost, while other knowledge that was discovered does not correctly



reflect reality. Furthermore, the model only considers the emergency medicine department of one hospital. The generalizability of the model is therefore low. Every hospital and/or department has their own capabilities, limitations and way of doing things.

During this case study, several ideas and opportunities came up to increase the success rate of similar projects in the future at the emergency medicine department of AZMM. The benefits of storing additional process-related information became more concrete, which further increases the willingness to make changes to the current situation. Some ideas entail changes to the underlying information system. For example, storing more timestamps concerning radiology and lab activities (i.e., time of request, time of execution, time when the results were made available, time when the physician looks at the results...). While other ideas involve to more radical changes to the way that the medical personnel go about their business. A medical scribe was made available during a test period to relieve the physicians of most of the writing activities in the EHR and at the same time to make the input more uniform. An expansion of the current capabilities to generate default texts based on shortcuts is also under consideration, possibly in combination with all options/answers so that the physician just needs to remove what is not applicable instead of having to type it. An even more ambitious idea of adding some sort of spelling correction to interface of the EHR was also discussed. All this extra structure in the textual input of the EHR will result in more precise data in the future, and consequently, more precise process models.

The method application described in this section required a substantial amount of manual work. This is typical for process mining projects, and even more so in healthcare settings (Rojas et al. 2016). The intrinsic complexity of the processes and the characteristic process-unaware information systems are the main culprits here. The long-term goal of the overarching research project and the emergency medicine department involved in the case is therefore, respectively, the development of and transition to a context- and process-aware information system. Such a system should understand the process and store the required data directly (e.g., an event log), which eliminates much of the manual work aside from the evaluation. From a research perspective, the manual work performed in the case confirms the need for such a system and provides a glimpse behind the scenes of a real-life loosely frame knowledge-intensive process. The healthcare organization on the other hand can view the manual work as a necessary investment to gather insight into the process taking place in order to facilitate the transition to a more process-aware information system in the future. But even when an organization is not eyeing a transition to a more process-aware information system, the amount of manual work can be regarded as an investment. Process mining projects are meant to be part of a BPM lifecycle in an organization (Dumas et al. 2018), which specifically states that it is a

cycle that needs to be repeated on a continuous basis. Thus, the manual work performed in the first iteration does not need to be redone each iteration, but rather, can be reused in the following iterations. As a result, the amount of manual work will be much lower in later iterations.

Finally, the proposed method should be regarded as a mere tool for process and decision discovery. With that, we mean that its success will depend heavily on the way it is applied. In Table 5, we provide a summary of the lessons we learned during this research and application project. These lessons are geared towards the data scientists who will be responsible for applying the method in future discovery projects. We have already touched upon some of these in the previous sections, but here we take a step back and discuss them on a project level.

## 6 Conclusion and Future Research

This paper describes the application of ADR to develop a method for process and decision discovery for loosely framed knowledge-intensive healthcare processes, centered around the DeciClareMiner tool. The method describes the steps to proceed from raw data in the IT-system of an organization to a detailed process and decision model of such a process. The use of a declarative process mining technique with integrated decision discovery is novel and requires modified approach compared to existing methods for process mining based on imperative process mining techniques. As part of the ADR-cycle, DeciClareMiner was applied for the first time to a real-life loosely framed knowledge-intensive healthcare process, namely the one that is performed in the emergency medicine department of a private hospital. This process is one of (if not the) most diverse, multi-disciplinary and dynamic processes in a hospital. Consequently, process and decision discovery in this setting is very challenging. Most of the process mining papers in a healthcare setting focus on a limited number of steps, aspects or pathologies (e.g., clinical pathways) to reduce the complexity of the task at hand. In contrast, the scope of this project was set out to be as comprehensive as possible with the available data. All recorded activities performed as part of the process in the emergency medicine department as well as all pathologies were considered from the viewpoint of the functional, control-flow and data perspectives of the process without resorting to filtering or clustering techniques to reduce the complexity.

The ‘spaghetti’ model that resulted from applying an imperative process mining technique to the data (Fig. 13) further strengthens the claim that imperative process modeling is not suitable for this type of processes. The understandability advantages (for human users) of imperative over declarative languages is completely negated by the high flexibility needs of the process and the resulting enumeration of all possible paths. Declarative languages are better suited due to their implicit

**Table 5** A summary of the lessons learned for data scientists

	Lesson	Description
1	<i>Start with a specific goal in mind</i>	It is important to define and discuss the goal(s) of the discovery project beforehand with the managers and physicians. The specific goals can have a significant impact on several steps of the method application. In this paper, we described two use cases with different goals that required different minimal confidence and support levels and evaluation. Other interesting use cases could focus on the physicians providing the treatment. For example, discovering a model that only includes patients that were treated by the more experienced physicians or a model per physician to compare them internally. Each of these use cases would require different implementations of the selection of data extracted from the information system and/or the event log filtering steps.
2	<i>The available data, data selection and data quality remain crucial</i>	<p>The process and decision discovery method is an important part of a discovery project. Yet, it is the available process data that will determine whether the project goals are achievable, even when everything else is done perfectly. The method will not produce any useful results without the right data of sufficiently high quality. This is essentially the 'garbage in, garbage out'-principle. Perfect data does not exist in real-life projects, so it is important to be aware of the imperfections and, where possible, try to mitigate them. Being aware of the data issues is also important to prevent wrong conclusions from being drawn from the results.</p> <p>Perhaps the most important role here is played by the domain expert(s) supporting the data scientists. Good domain experts have experiences on the work floor and are preferably still active there. As a data scientist, it is essential to thoroughly explain what you need and what the eventual goals are to the domain experts. If they fully understand those, they can anticipate your needs and point you in the right direction. Also involve them as much as possible in the data selection and quality review. They can give valuable insight into what is being inputted into the system and how trustworthy this data is.</p> <p>In our method application, we had to deal with multiple data problems such as data not being registered, not having access to certain useful data (e.g., no patient age or gender due to patient privacy regulations), pruning out unrelated or low quality data attributes and managing different types of noise. Therefore, we performed a critical review of the available data with together with people from different levels of the organization:</p> <ul style="list-style-type: none"> <li>- After a thorough discussion of the purpose of the discovery method, we defined the project with the head of the emergency department. This included a review of the information system as used on the work floor to get a good feel for what is being recorded and the subsequent opportunities for the potential goals of the project. This provided the foundation for the data selection, although at this point still in terms of data inputs of the information system.</li> <li>- The IT-department provided a mapping of the data inputs in the information system to the model of the extracted data. This revealed the extent to which the requested data was scattered across multiple databases and some other problems for which we had to devise mitigation strategies.</li> <li>- The physicians were asked to explain how they made use of certain data input fields in the information system. This uncovered some misalignments between how the intended usage by the IT-department and the actual usage by the physicians, as well as some creative use to get around certain limitations. Based on this information, we could prune out certain low-quality data attributes and implement several data quality improvement steps.</li> <li>- Together with the head of the emergency department, who was also our domain expert, we subsequently ran through many process instances during the evaluation step to review both the quality of the data and of the model based on that data. This exposed additional data issues, like registered data that was not provided to us or misinterpreted data. Most of these issues could be solved during a subsequent iteration of the method.</li> </ul> <p>The remaining data and quality issues were documented during the method application and linked to certain parts of the discovered model to make the eventual users of the model aware of its weaknesses.</p>
3	<i>Favor hierarchical activities over branching</i>	The underlying process logic often involves more than the minimal number of activities linked to a certain declarative rule. Increasing the branching level will allow the miner to discover more of this logic. However, the percentage of spurious relationships in the results will likely rise and the added computational effort is very high. When possible, hierarchical activities should be defined to group some lower level activities or even other hierarchical activities. This is a manual step that requires some domain knowledge but could certainly pay dividends later down the line. Mining hierarchical activities is much more computationally efficient and prevents many of the spurious relationships to be discovered. If the domain knowledge is not available to define the hierarchical activities, one can still resort to higher branching levels when the sufficient computational power is available.

**Table 5** (continued)

Lesson	Description
4 <i>The identification of data elements is a tough balancing exercise</i>	<p>A naive approach to the identification of data elements might be ‘the more the merrier’. After all, the discovery of decision logic depends on having the necessary data elements in the event log. Yet, the practical reality is that this is a difficult trade-off between potentially uncovering more of the real decision logic versus the general quality of the discovered decision logic and the computational effort. Lower quality data elements might lower the quality of the mined model because they could be incorrectly linked to certain discovered decisions, while they also significantly increase the search space leading to a lot of waste mining time. Therefore, the identification and selection of data elements to use in the event log should be done with much care.</p> <p>We would advise to start from a small set of potential data element candidates, manually review the quality as much as possible and iteratively increase/refine the data elements throughout the project based on the results of the evaluation step at the end of each iteration.</p>
5 <i>Test multiple parameter combinations in parallel</i>	<p>The selection of the maximal branching level and minimal confidence and support levels can be a bit of trial and error. It is not wrong to choosing one specific combination of values for these parameters for each iteration, but it can be inefficient. By mining with multiple parameter combination per iteration, comparisons can be made of the results early in the project. The different models can be evaluated in the same evaluation step at the end of each iteration to save some time.</p> <p>Also, keep in mind that the mining step is not deterministic nor complete. Parameter combinations that range from general (high support) to specific (low support) and strict (high confidence) to more lenient (low confidence) can be chosen so that the more general and strict parameter settings are subsumed by each subsequent parameter setting with less general and more lenient levels. So, all models that are more general and stricter can be merged directly in the other models. In general, all the rules in the discovered models can be combined and new models of each parameter combination can be created by filtering the rules with the specific thresholds. The resulting models will be more complete because the search space directly or indirectly explored in these models will be at least equal to, but almost always bigger than, the search space explored during each of the original mining steps. This means that the additional computational effort required to mine more than one model per iteration for internal comparison is optimally leveraged to increase the quality of all models.</p>

incorporation of process variations. The combination of this declarative viewpoint and an extensive data perspective enables DeciClare models to capture both the loosely framed and the knowledge-intensive characteristics of the targeted processes. As a result, the discovered DeciClare model of the emergency care process offers a glimpse into the previously tacit knowledge about how patients are diagnosed and treated at the emergency medicine department of the hospital. Transforming this tacit knowledge to explicit knowledge can have many benefits to the department, the hospital and possibly even to the medical field in general (e.g., transparency, analysis, comparison, optimization, education...). The evaluation demonstrates that realistic functional, control-flow and decision logic can be identified with the developed method.

This research contributes to the information systems community by bringing together the state-of-the-art from different fields like computer science, business process management, conceptual modeling, process mining, decision management and knowledge representation to make advancements in the field of medical informatics. Although many advancements have recently been made in process mining research, most of the proposed techniques and methods are either geared

towards tightly framed processes or lack support for the knowledge governing the path decisions. We have addressed this research gap for the discovery of loosely framed knowledge-intensive healthcare processes. The proposed method is not merely a theoretical contribution, it is a practical contribution. This was demonstrated by the application of the method to a real-life example process, which both shows its feasibility and its potential. Data scientists can use this as a template to create similar models in other healthcare organizations. We envision the method and corresponding tools to be integrated into the hospital information systems of the future to allow for the automatic discovery of such models directly. This could become part of the review process that many healthcare organizations already undergo periodically to verify that clinical guidelines and hospital policies are being applied correctly. On the other hand, the model itself enables the use of process engines like DeciClareEngine that offer automatic interpretation of the model and support users while executing the process by providing them with the right pieces of knowledge at the right time. This is a necessary step towards offering real-time process and decision support covering the complete process, as opposed to current generation of clinical

decision support systems that only offer support at some very specific decision point(s) (e.g., for the prevention venous thromboembolism (Durieux et al. 2000)).

As future research, we need to make a distinction between the developed method and the practical case through which the method was developed. In the context of the method, we will further refine it by applying it to other healthcare process, while following the recommendations of Martin et al. (Martin et al. 2020). We will also verify its general applicability to other loosely framed knowledge-intensive processes by applying it to datasets from other domains. From the perspective of the practical case, a project like this does not really have a definite ending. It will need to be repeated continuously as initiatives to improve the data quality start to bear fruit. The quality of the discovered process and decision model will improve with more and better data, which in turn externalizes more and more real-life medical knowledge. Additionally, we are investigating how the process and decision model can be used to help users during the execution of the process. The process engine used in the evaluation of the discovered model is a first step in this direction, yet the development of a true context- and process-aware information system is still ongoing.

**Acknowledgments** This work has been partially supported by the Flemish Agency for Innovation and Entrepreneurship (VLAIO) as part of project HBC.2019.2223. We also would like to thank the AZ Maria Middelares hospital in Ghent, Belgium, for their support and cooperation.

## References

- van der Aalst, W. M. P. (2011). Process mining. In: *Discovery, conformance and enhancement of business processes*. :Springer. <https://doi.org/10.1007/978-3-642-19345-3>.
- van der Aalst, W. M. P. (2013). Business process management: A comprehensive survey. In: *ISRN Software Engineering*, pp. 1–37. <https://doi.org/10.1155/2013/507984>.
- van der Aalst, W. M. P., Pesic, M., & Schonenberg, H. (2009). Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development*, 23(2), 99–113. <https://doi.org/10.1007/s00450-009-0057-9>.
- Abo-Hamad, W. (2017). “Patient pathways discovery and analysis using process mining techniques: an emergency department case study,” in *Health Care Systems Engineering (ICHSE)*. Springer Proceedings in Mathematics and Statistics. PROMS Vol. 210. :Springer, pp. 209–219. [https://doi.org/10.1007/978-3-319-66146-9\\_19](https://doi.org/10.1007/978-3-319-66146-9_19).
- Agrawal, R., and Srikant, R. (1994). “Fast algorithms for mining association rules,” in *VLDB*, pp. 487–499. <https://doi.org/10.1.1.40.7506>.
- Andersen, S. N., & Broberg, O. (2017). A framework of knowledge creation processes in participatory simulation of hospital work systems. *Ergonomics*, 60(4), 487–503. <https://doi.org/10.1080/00140139.2016.1212999>.
- Antunes, P., Pino, J. A., Tate, M., & Barros, A. (2020). Eliciting process knowledge through process stories. *Information Systems Frontiers*, 22, 1179–1201. <https://doi.org/10.1007/s10796-019-09922-0>.
- Baker, K., Dunwoodie, E., Jones, R. G., Newsham, A., Johnson, O., Price, C. P., Wolstenholme, J., Leal, J., McGinley, P., Twelves, C., & Hall, G. (2017). Process mining routinely collected electronic health records to define real-life clinical pathways during chemotherapy. *International Journal of Medical Informatics*, 103, 32–41. <https://doi.org/10.1016/j.ijmedinf.2017.03.011> Elsevier Ireland Ltd.
- Basole, R. C., Braunstein, M. L., Kumar, V., Park, H., Kahng, M., Chau, D. H., Tamersoy, A., Hirsh, D. A., Serban, N., Bost, J., Lesnick, B., Schissel, B. L., & Thompson, M. (2015). Understanding variations in pediatric asthma care processes in the emergency department using visual analytics. *Journal of the American Medical Informatics Association*, 22(2), 318–323. <https://doi.org/10.1093/jamia/ocu016>.
- De Bleser, L., Depreitere, R., De Waele, K., Vanhaecht, K., Vlayen, J., & Sermeus, W. (2006). Defining pathways. *Journal of Nursing Management*, 14(7), 553–563. <https://doi.org/10.1111/j.1365-2934.2006.00702.x>.
- Braga, B. F. B., Almeida, J. P. A., Guizzardi, G., & Benevides, A. B. (2010). Transforming OntoUML into alloy: Towards conceptual model validation using a lightweight formal method. *Innovations in Systems and Software Engineering*, 6(1), 55–63. <https://doi.org/10.1007/s11334-009-0120-5>.
- Burattin, A., Cimitile, M., Maggi, F. M., & Sperduti, A. (2015). Online discovery of declarative process models from event streams. *IEEE Transactions on Services Computing*, 8(6), 833–846. <https://doi.org/10.1109/TSC.2015.2459703> IEEE.
- Burattin, A., Maggi, F. M., & Sperduti, A. (2016). Conformance checking based on multi-perspective declarative process models. *Expert Systems with Applications*, 65, 194–211. <https://doi.org/10.1016/j.eswa.2016.08.040> Elsevier Ltd.
- Bygstad, B., Øvrelid, E., Lie, T., & Bergquist, M. (2020). Developing and organizing an analytics capability for patient flow in a general hospital. *Information Systems Frontiers*, 22, 353–364. <https://doi.org/10.1007/s10796-019-09920-2>.
- Campbell, H., Hotchkiss, R., Bradshaw, N., & Porteous, M. (1998). Integrated care pathways. *BMJ*, 316, 133. <https://doi.org/10.1136/bmj.316.7125.133>.
- Di Ciccio, C., Maggi, F. M., Montali, M., & Mendling, J. (2017). Resolving Inconsistencies and redundancies in declarative process models. *Information Systems*, 64, Elsevier Ltd, 425–446. <https://doi.org/10.1016/j.is.2016.09.005>.
- Di Ciccio, C., Marrella, A., & Russo, A. (2015a). Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches. *Journal on Data Semantics*, 4(1), 29–57. <https://doi.org/10.1007/s13740-014-0038-4>.
- Di Ciccio, C., Mecella, M., and Mendling, J. (2015b). “The effect of noise on mined declarative constraints,” in *Data-driven process discovery and analysis (SIMPDA’13)*. LNBP Vol. 203. :Springer, pp. 1–24. [https://doi.org/10.1007/978-3-662-46436-6\\_1](https://doi.org/10.1007/978-3-662-46436-6_1).
- Cognini, R., Corradini, F., Gnesi, S., Polini, A., & Re, B. (2018). Business process flexibility - a systematic literature review with a software systems perspective. *Information Systems Frontiers*, 20(2), 343–371. <https://doi.org/10.1007/s10796-016-9678-2>.
- Combi, C., Oliboni, B., and Gabrieli, A. (2015). “Conceptual modeling of clinical pathways: Making data and processes connected,” in *15th Conference on Artificial Intelligence in Medicine (AIME)*, Pavia, Italy pp. 57–62. [https://doi.org/10.1007/978-3-319-19551-3\\_7](https://doi.org/10.1007/978-3-319-19551-3_7).
- Daelemans, W., Zavrel, J., van Der Sloot, K., and van Den Bosch, A. (2009). TiMBL: Tilburg memory based learner, version 6.2, reference guide, ILK Research Group Technical Report Series (07–07), p. 66.
- Deokar, A. V., & El-Gayar, O. F. (2011). Decision-enabled dynamic process management for networked enterprises. *Information Systems Frontiers*, 13, 655–668. <https://doi.org/10.1007/s10796-010-9243-3>.
- Dijkman, R., Vincent, S., & Ad, L. (2016). Properties that influence business process management maturity and its effect on



- organizational performance. *Information Systems Frontiers*, 18, 717–734. <https://doi.org/10.1007/s10796-015-9554-5>.
- Duma, D., & Aringhieri, R. (2020). An ad hoc process mining approach to discover patient paths of an emergency department. *Flexible Services and Manufacturing Journal*, 32, 6–34. <https://doi.org/10.1007/s10696-018-9330-1> Springer.
- Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of business process management* (2nd ed.). Berlin: Springer-Verlag. <https://doi.org/10.1007/978-3-662-56509-4>.
- Durieux, P., Nizard, R., Ravaud, P., Mounier, N., & Lepage, E. (2000). A clinical decision support system for prevention of venous thromboembolism. *JAMA*, 283(21), 2816–2821. <https://doi.org/10.1001/jama.283.21.2816>.
- van Eck, M. L., Lu, X., Leemans, S. J. J., & van der Aalst, W. M. P. (2015). PM<sup>2</sup>: a process mining project methodology. In J. Zdravkovic, M. Kirikova, & P. Johannesson (Eds.), *CAiSE'15. LNCS, Vol. 9097* (pp. 297–313). [https://doi.org/10.1007/978-3-319-19069-3\\_19](https://doi.org/10.1007/978-3-319-19069-3_19).
- Fernandez-Llatas, C., Martinez-Millana, A., Martinez-Romero, A., Benedi, J. M., and Traver, V. (2015). “Diabetes care related process modelling using process mining techniques. Lessons learned in the application of interactive pattern recognition: coping with the spaghetti effect,” in 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE Xplore, pp. 2127–2130. <https://doi.org/10.1109/EMBC.2015.7318809>.
- Funkner, A. A., Yakovlev, A. N., & Kovalchuk, S. V. (2017). Towards evolutionary discovery of typical clinical pathways in electronic health records. *Procedia Computer Science*, 119, 234–244. <https://doi.org/10.1016/j.procs.2017.11.181> Elsevier.
- Ghasemi, M., and Amyot, D. 2016. “Process mining in healthcare: A systematised literature review,” *International Journal of Electronic Healthcare* (9:1). <https://doi.org/10.1504/IJEH.2016.078745>.
- Goedertier, S., Vanthienen, J., & Caron, F. (2015). Declarative business process modelling: Principles and modelling languages. *Enterprise Information Systems*, 9(2), 161–185. <https://doi.org/10.1080/17517575.2013.830340>.
- Greenes, R. A., Bates, D. W., Kawamoto, K., Middleton, B., Osheroff, J., & Shahar, Y. (2018). Clinical decision support models and frameworks: Seeking to address research issues underlying implementation successes and failures. *Journal of Biomedical Informatics*, 78, 134–143. <https://doi.org/10.1016/j.jbi.2017.12.005>.
- Guizzardi, G., Wagner, G., De Almeida Falbo, R., Guizzardi, R. S. S., and Almeida, J. P. A. 2013. “Towards Ontological Foundations for the Conceptual Modeling of Events,” in *International Conference on Conceptual Modeling (ER)*. LNCS Vol. 8217, W. Ng, V. C. Storey, and J. C. Trujillo (eds.), Springer, pp. 327–341. [https://doi.org/10.1007/978-3-642-41924-9\\_27](https://doi.org/10.1007/978-3-642-41924-9_27).
- Haisjackl, C., Barba, I., Zugal, S., Soffer, P., Hadar, I., Reichert, M., Pinggera, J., and Weber, B. 2014. “Understanding Declare Models: Strategies, Pitfalls, Empirical Results,” *Software & Systems Modeling* (Special Se), pp. 1–28. <https://doi.org/10.1007/s10270-014-0435-z>.
- Hay, M., Weisner, T. S., Subramanian, S., Duan, N., Niedzinski, E. J., & Kravitz, R. L. (2008). Harnessing experience: Exploring the gap between evidence-based medicine and clinical practice. *Journal of Evaluation in Clinical Practice*, 14(5), 707–713. <https://doi.org/10.1111/j.1365-2753.2008.01009.x>.
- Helm, E., & Paster, F. (2015). First steps towards process Mining in Distributed Health Information Systems. *International Journal of Electronics and Telecommunications*, 61(2), 137–142. <https://doi.org/10.1515/eletel-2015-0017>.
- Hertz, S., Johansson, J. K., & de Jager, F. (2001). Customer-oriented cost cutting: Process Management at Volvo. *Supply Chain Management*, 6(3), 128–142. <https://doi.org/10.1108/13598540110399174>.
- Heß, M., Kaczmarek, M., Frank, U., Podleska, L.-E., and Taeger, G. 2015. “Towards a Pathway-Based Clinical Cancer Registration in Hospital Information Systems,” in *Knowledge Representation for Health Care. LNCS Vol. 9485*, Springer, pp. 80–94. [https://doi.org/10.1007/978-3-319-26585-8\\_6](https://doi.org/10.1007/978-3-319-26585-8_6).
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.
- Hildebrandt, T., Mukkamala, R. R., & Slaats, T. (2012). Declarative Modelling and safe distribution of healthcare workflows. In Z. Liu & A. Wassung (Eds.), *FHIES 2011. LNCS Vol. 7151* (pp. 39–56). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-32355-3\\_3](https://doi.org/10.1007/978-3-642-32355-3_3).
- Huang, Z., Bao, Y., Dong, W., Lu, X., and Duan, H. 2014. “Online Treatment Compliance Checking for Clinical Pathways,” *Journal of Medical Systems* (38:10). <https://doi.org/10.1007/s10916-014-0123-0>.
- Huang, Z., Dong, W., Ji, L., He, C., and Duan, H. 2016. “Incorporating Comorbidities into Latent Treatment Pattern Mining for Clinical Pathways,” *Journal of Biomedical Informatics* (59), Elsevier Inc., pp. 227–239. <https://doi.org/10.1016/j.jbi.2015.12.012>.
- Kamsu-Foguem, B., Tchuenté-Foguem, G., & Foguem, C. (2014). Using conceptual graphs for clinical guidelines representation and knowledge visualization. *Information Systems Frontiers*, 16(4), 571–589. <https://doi.org/10.1007/s10796-012-9360-2>.
- Kim, G., & Suh, Y. (2011). Semantic business process space for intelligent Management of Sales Order Business Processes. *Information Systems Frontiers*, 13(4), 515–542. <https://doi.org/10.1007/s10796-010-9229-1>.
- Kimble, C., de Vasconcelos, J. B., & Rocha, Á. (2016). Competence Management in Knowledge Intensive Organizations Using Consensual Knowledge and Ontologies. *Information Systems Frontiers*, 18(6), 1119–1130. <https://doi.org/10.1007/s10796-016-9627-0>.
- Klein, J. H., and Young, T. 2015. “Health Care: A Case of Hypercomplexity?,” *Health Systems* (4:2), Springer, pp. 104–110. <https://doi.org/10.1057/hs.2014.21>, Health care: a case of hypercomplexity?.
- Kluza, K., & Nalepa, G. J. (2019). Formal model of business processes integrated with business rules. *Information Systems Frontiers*, 21(5), 1167–1185. <https://doi.org/10.1007/s10796-018-9826-y>.
- Kovalchuk, S., Funkner, A., Metsker, O., and Yakovlev, A. (2018). “Simulation of Patient Flow in Multiple Healthcare Units Using Process and Data Mining Techniques for Model Identification,” *Journal of Biomedical Informatics* (82), Elsevier, pp. 128–142. <https://doi.org/10.1016/j.jbi.2018.05.004>.
- Krishnan, M. S., Mukhopadhyay, T., & Zubrow, D. (1999). Software process models and project performance. *Information Systems Frontiers*, 1, 267–277. <https://doi.org/10.1023/A:1010054412650>.
- Kurniati, A. P., Johnson, O., Hogg, D., and Hall, G. 2016. “Process Mining in Oncology: A Literature Review,” in *ICICM 2016*, IEEE, October, pp. 291–297. <https://doi.org/10.1109/INFOCOMAN.2016.7784260>.
- Lawal, A. K., Rotter, T., Kinsman, L., Machotta, A., Ronellenfisch, U., Scott, S. D., Goodridge, D., Plishka, C., and Groot, G. (2016). “What Is a Clinical Pathway? Refinement of an Operational Definition to Identify Clinical Pathway Studies for a Cochrane Systematic Review,” *BMC Medicine* (14:1), BMC Medicine, pp. 1–5. <https://doi.org/10.1186/s12916-016-0580-z>.
- Lenz, R., & Reichert, M. (2007). IT support for healthcare processes - premises, challenges, perspectives. *Data & Knowledge Engineering*, 61(1), 39–58. <https://doi.org/10.1016/j.datak.2006.04.007>.
- Maggi, F. M., Dumas, M., García-Bañuelos, L., and Montali, M. 2013. “Discovering Data-Aware Declarative Process Models from Event Logs,” in *BPM'13. LNCS Vol. 8094*, F. Daniel, J. Wang, and B.

- Weber (eds.), Springer, pp. 81–96. [https://doi.org/10.1007/978-3-642-40176-3\\_8](https://doi.org/10.1007/978-3-642-40176-3_8).
- Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.
- Mardini, M. T., & Ras, Z. W. (2020). Discovering primary medical procedures and their associations with other procedures in HCUP data. *Information Systems Frontiers*. <https://doi.org/10.1007/s10796-020-10058-9>.
- Mariscal, G., Marbán, Ó., & Fernández, C. (2010). A survey of data mining and knowledge discovery process models and methodologies. *Knowledge Engineering Review*, 25(2), 137–166. <https://doi.org/10.1017/S0269888910000032>.
- Martin, N., De Weerd, J., Fernandez-Llatas, C., Gal, A., Gatta, R., Ibáñez, G., Johnson, O., Mannhardt, F., Marco-Ruiz, L., Mertens, S., Munoz-Gama, J., Seoane, F., Vanthienen, J., Wynn, M. T., Boilève, D. B., Bergs, J., Joosten-Melis, M., Schretlen, S., and Van Acker, B. 2020. “Recommendations for Enhancing the Usability and Understandability of Process Mining in Healthcare,” *Artificial Intelligence in Medicine* (109). <https://doi.org/10.1016/j.artmed.2020.101962>.
- Mertens, S., Gailly, F., and Poels, G. 2017. “Towards a Decision-Aware Declarative Process Modeling Language for Knowledge-Intensive Processes,” *Expert Systems with Applications* (87), Elsevier Ltd, pp. 316–334. <https://doi.org/10.1016/j.eswa.2017.06.024>.
- Mertens, S., Gailly, F., and Poels, G. 2018. “Discovering Health-Care Processes Using DeciClareMiner,” *Health Systems - Special Issue: SIG-Health Design, Innovation, and Impact of Healthcare IT* (7:3), Taylor & Francis, pp. 195–211. <https://doi.org/10.1080/20476965.2017.1405876>.
- Mertens, S., Gailly, F., and Poels, G. 2019a. “A generic framework for flexible and data-aware business process engines,” in *Advanced information systems engineering (CAiSE) Workshops. LNBIP Vol. 349*, H. Proper and J. Stima (eds.), Springer, Cham, pp. 201–213. [https://doi.org/10.1007/978-3-030-20948-3\\_18](https://doi.org/10.1007/978-3-030-20948-3_18).
- Mertens, S., Gailly, F., Van Sassenbroeck, D., and Poels, G. (2019b). “Comparing Strategies to Generate Experience-Based Clinical Process Recommendations That Leverage Similarity to Historic Data,” in *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, Xi’an, China: IEEE. <https://doi.org/10.1109/ICHI.2019.8904693>.
- Nunes, V. T., Santoro, F. M., Werner, C. M. L., & Ralha, C. G. (2016). Context and planning for dynamic adaptation in PAIS. In M. Reichert & H. A. Reijers (Eds.), *BPM workshops. LNBIP Vol. 256* (pp. 471–483). Cham: Springer. [https://doi.org/10.1007/978-3-319-42887-1\\_38](https://doi.org/10.1007/978-3-319-42887-1_38).
- Orellana Garcia, A., Perez Ramirez, Y. E., & Armenteros Larrea, O. U. (2015). Process Mining in Healthcare: Analysis and modeling of processes in the emergency area. *IEEE Latin America Transactions*, 13(5), 1612–1618. <https://doi.org/10.1109/TLA.2015.7112022>.
- Palvia, P., Lowe, K., Nemati, H., and Jacks, T. 2012. “Information Technology Issues in Healthcare: Hospital CEO and CIO Perspectives,” *Information Technology* (30). [http://libres.uncg.edu/ir/uncg/f/K\\_Lowe\\_Information\\_2012.pdf](http://libres.uncg.edu/ir/uncg/f/K_Lowe_Information_2012.pdf).
- Payton, F. C., Paré, G., LeRouge, C., and Reddy, M. 2011. “Health Care IT: Process, People, Patients and Interdisciplinary Considerations,” *Journal of the Association for Information Systems* (12:2). <https://doi.org/10.17705/1jais.00259>.
- Pesic, M. (2008). *Constraint-based workflow management systems: Shifting control to users*. Doctoral Dissertation: Technische Universiteit Eindhoven. <https://doi.org/http://www.narcis.nl/publication/RecordID/oai:library.tue.nl:638413>.
- Rebuge, Á., & Ferreira, D. R. (2012). Business process analysis in healthcare environments: A methodology based on process mining. *Information Systems*, 37(2), 99–116. <https://doi.org/10.1016/j.is.2011.01.003>.
- Rehse, J.-R., Fettke, P., and Loos, P. 2018. “Process Mining and the Black Swan: An Empirical Analysis of the Influence of Unobserved Behavior on the Quality of Mined Process Models,” in *BPM Workshops. LNBIP Vol 308*, E. Teniente and M. Weidlich (eds.), Springer, pp. 256–268. [https://doi.org/10.1007/978-3-319-74030-0\\_19](https://doi.org/10.1007/978-3-319-74030-0_19).
- Rojas, E., Arias, M., and Sepúlveda, M. 2015. “Clinical Processes and Its Data, What Can We Do with Them?,” in *HEALTHINF '15, SCITEPRESS - Science and Technology Publications*, pp. 642–647. <https://doi.org/10.5220/0005287206420647>.
- Rojas, E., Cifuentes, A., Burattin, A., Munoz-Gama, J., Sepúlveda, M., and Capurro, D. 2019. “Performance Analysis of Emergency Room Episodes through Process Mining,” *International Journal of Environmental Research and Public Health* (16:7). <https://doi.org/10.3390/ijerph16071274>.
- Rojas, E., Munoz-Gama, J., Sepúlveda, M., and Capurro, D. 2016. “Process Mining in Healthcare: A Literature Review,” *Journal of Biomedical Informatics* (61), Elsevier Inc., pp. 224–236. <https://doi.org/10.1016/j.jbi.2016.04.007>.
- Rotter, T., Kinsman, L., James, E. L., Machotta, A., Gothe, H., Willis, J., Snow, P., and Kugler, J. 2010. “Clinical Pathways: Effects on Professional Practice, Patient Outcomes, Length of Stay and Hospital Costs,” *Cochrane Database of Systematic Reviews* (3), (T. Rotter, ed.), Chichester, UK: John Wiley & Sons, Ltd, pp. 1–170. <https://doi.org/10.1002/14651858.CD006632.pub2>.
- Rovani, M., Maggi, F. M., de Leoni, M., and van der Aalst, W. M. P. 2015. “Declarative Process Mining in Healthcare,” *Expert Systems with Applications* (42:23), Elsevier Ltd., pp. 9236–9251. <https://doi.org/10.1016/j.eswa.2015.07.040>.
- Santoro, F. M., Slaats, T., Hildebrandt, T. T., and Baião, F. A. 2019. “DCR-KiPN a hybrid modeling approach for knowledge-intensive processes,” in *2019 Conceptual Modeling (ER). LNCS Vol. 11788*, A. Laender, B. Pernici, E. Lim, and J. de Oliveira (eds.), Springer, Cham, pp. 153–161. [https://doi.org/10.1007/978-3-030-33223-5\\_13](https://doi.org/10.1007/978-3-030-33223-5_13).
- Schönig, S., Di Ciccio, C., Maggi, F. M., and Mendling, J. 2016. “Discovery of Multi-Perspective Declarative Process Models,” in *ICSOC 2016: Service-Oriented Computing. LNCS Vol. 9936*, Q. Sheng, E. Stroulia, S. Tata, and S. Bhiri (eds.), Springer, pp. 87–103. [https://doi.org/10.1007/978-3-319-46295-0\\_6](https://doi.org/10.1007/978-3-319-46295-0_6).
- Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action design research. *MIS Quarterly*, 35(1), 37–56. <https://doi.org/10.2307/23043488>.
- Shearer, C. (2000). The crisp-Dm model: The new blueprint for data mining. *Journal of Data Warehousing*, 5(4), 13–22.
- Sittig, D. F., Wright, A., Osheroff, J. A., Middleton, B., Teich, J. M., Ash, J. S., Campbell, E., & Bates, D. W. (2008). Grand challenges in clinical decision support. *Journal of Biomedical Informatics*, 41(2), 387–392. <https://doi.org/10.1016/j.jbi.2007.09.003>.
- van der Spoel, S., van Keulen, M., and Amrit, C. 2013. “Process Prediction in Noisy Data Sets: A Case Study in a Dutch Hospital,” in *SIMPDA'12. LNBIP Vol. 162*, P. Cudre-Mauroux, P. Ceravolo, and D. Gašević (eds.), pp. 60–83. [https://doi.org/10.1007/978-3-642-40919-6\\_4](https://doi.org/10.1007/978-3-642-40919-6_4).
- Stefanelli, M. (2004). Knowledge and process Management in Health Care Organizations. *Methods of Information in Medicine*, 43(05), 525–535. <https://doi.org/10.1055/s-0038-1633911>.
- Suriadi, S., Andrews, R., ter Hofstede, A. H. M., and Wynn, M. T. 2017. “Event Log Imperfection Patterns for Process Mining: Towards a Systematic Approach to Cleaning Event Logs,” *Information Systems* (64), Elsevier, pp. 132–150. <https://doi.org/10.1016/j.is.2016.07.011>.
- Vanbrabant, L., Martin, N., Ramaekers, K., and Braekers, K. 2019. “Quality of Input Data in Emergency Department Simulations: Framework and Assessment Techniques,” *Simulation Modelling*

- Practice and Theory* (91), Elsevier, pp. 83–101. <https://doi.org/10.1016/j.simpat.2018.12.002>.
- Vanhaecht, K., De Witte, K., Depreitere, R., & Sermeus, W. (2006). Clinical pathway audit tools: A systematic review. *Journal of Nursing Management*, 14(7), 529–537. <https://doi.org/10.1111/j.1365-2934.2006.00705.x>.
- Wang, T., Tian, X., Yu, M., Qi, X., & Yang, L. (2017). Stage division and pattern discovery of complex patient care processes. *Journal of Systems Science and Complexity*, 30(5), 1136–1159. <https://doi.org/10.1007/s11424-017-5302-x>.
- Wyatt, J. C. (2001). Management of Explicit and Tacit Knowledge. *Journal of the Royal Society of Medicine*, 94(1), 6–9. <https://doi.org/10.1177/014107680109400102>.
- Zhang, Y., Padman, R., and Patel, N. 2015. “Paving the COWpath: Learning and Visualizing Clinical Pathways from Electronic Health Record Data,” *Journal of Biomedical Informatics* (58), Elsevier, pp. 186–197. <https://doi.org/10.1016/j.jbi.2015.09.009>.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Steven Mertens** is a postdoctoral researcher for the department of Business Informatics and Operations Management at Ghent University, Belgium. After his civil engineer studies at the KU Leuven (Belgium), he received his PhD from Ghent University in 2019 for his work on the subject of process support for loosely framed knowledge-intensive processes. He received the CAiSE Best PhD award in 2020, while continuing his research on a personal research grant from the Flemish Agency for Innovation and Entrepreneurship (VLAIO).

**Frederik Gailly** is an associate professor at Ghent University. He is member of the UGent Business Informatic research group, which is part of the

Faculty of Economics and Business Administration of Ghent University. His research focuses on two relevant information systems research areas. Firstly, his research focuses on ontologies and on how they can be applied in the context of conceptual modelling in general and process modelling more specific. Secondly, his research focuses on the development of methods, techniques and tools that can be used to discover, model, and execute knowledge intensive business processes.

**Diederik Van Sassenbroeck** is currently the head of the emergency department and medical care manager in the AZ Maria Middelaere hospital in Ghent, Belgium. Dr. Diederik Van Sassenbroeck was trained in internal medicine and emergency medicine at Ghent University, Belgium. As a PhD student at the Heymans Institute for pharmacology in Ghent, he was trained in pharmacokineticpharmacodynamic modelling. While working as a medical care manager, he later developed capacity models for hospital functions like emergency departments and hospital wards towards his additional master's in Hospital Management and Health Policy from the University of Leuven, Belgium. He was trained to be, and certified as, an emergency department executive by the International Emergency Department Leadership Institute of the Brigham and Women university hospital of the Harvard University in Boston, USA. Nowadays he is interests include the rational approach in diagnosis and treatment of populations rather than the care for single patients.

**Geert Poels** is a senior full professor in Management Information Systems at the Faculty of Economics and Business Administration of Ghent University, Belgium. He directs the Business Informatics of the Department of Business Informatics and Operations Management and coordinates the Enterprise Architecture research cluster. His research interests include conceptual modelling, business process architecture, and service science. He also teaches, guides master dissertation research, and is a program board member of the Master in Enterprise Architecture organized by IC Institute, Beerzel, Belgium.