From BPMN Process Models to DMN Decision Models

Ekaterina Bazhenova *[a], Francesca Zerbato *[b], Barbara Oliboni[b], Mathias Weske[a]

[a] *Hasso Plattner Institute, University of Potsdam, Germany*
[b] *Department of Computer Science, University of Verona, Italy*

Abstract

The interplay between process and decision models plays a crucial role in business process management, as decisions may be based on running processes and affect process outcomes. Often process models include decisions that are encoded through process control flow structures and data flow elements, thus reducing process model maintainability. The Decision Model and Notation (DMN) was proposed to achieve separation of concerns and to possibly complement the Business Process Model and Notation (BPMN) for designing decisions related to process models. Nevertheless, deriving decision models from process models remains challenging, especially when the same data underlie both process and decision models. In this paper, we explore how and to which extent the data modeled in BPMN processes and used for decision-making may be represented in the corresponding DMN decision models. To this end, we identify a set of patterns that capture possible representations of data in BPMN processes and that can be used to guide the derivation of decision models related to existing process models. Throughout the paper we refer to real-world healthcare processes to show the applicability of the proposed approach.

Keywords: Business process models, decision models, BPMN, DMN

1. Introduction

In the field of business process management (BPM), decisions are becoming increasingly integrated into business processes [1], as they boost organizational competitiveness and may be analyzed, implemented, and reused in multiple processes for improving business outcomes [2]. However, since both BPM and decision management have existed long without proper integration, process modeling

*Both authors contributed equally to this manuscript.

languages have often been misused for designing decisions [3], thus resulting in complex "spaghetti" models [4] that are hard to read and to maintain [5].

To overcome this limitation, the Decision Model and Notation (DMN) [6] was developed by the Object Management Group for modeling decisions at different levels of detail. One of the main aims of the DMN standard is to complement the Business Process Model and Notation (BPMN 2.0) [7] for decision design, thus contributing to create a standard-based framework to support the design of processes and related decisions.

In detail, the DMN standard provides a complete two-layered decision model that combines decision requirements, which are represented through suitable diagrams, and decision logic. One of the ways lo link DMN decision models to BPMN process models is by associating decisions with those process activities within which the decision-making takes place [6]. Such *decision activities* may be linked to a decision model that details the decision requirements and the inner decision logic of the activity.

The combination of BPMN and DMN models allows one to naturally model decision logic separately from process logic, thus achieving a "separation of concerns" [8], which is eases process and decision model maintainability [5, 9]. When decisions are defined in process structures, any modification of the decision logic needs to be reflected in the process model. Thus, separating concerns is worthwhile, especially when process and decision models are maintained and re-engineered by different stakeholders.

In this direction, the need for a separate, yet integrated modeling of decisions and processes has become central in BPM research [9]. In organizational realities, process and decision models are closely interrelated, as decisions may drive the process flow and processes may manipulate information that is used to make decisions. In particular, decision activities take in input data created or acquired earlier in the process, and produce decision outputs which may be used later in the process. Whereas separating concerns is easy for newly modeled processes, separating concerns in existing processes becomes quite challenging, especially when decision-making aspects are integrated within process models, and both process and decision models rely on the same, shared business data.

In this paper, we consider the extraction of DMN decision models from BPMN process models [3, 4, 5], focusing on the data perspective of process models and providing an approach to derive a DMN model including such decisions.

Previous work tackled the discovery of DMN models from the process control flow [3, 5], or from event logs [10, 11]. However, how to unbundle decisions captured by the data perspective of BPMN process models is yet to be investigated.

The main contribution of this paper is to provide a pattern-based approach to support decision designers and analysts in understanding how the data explicitly represented in a process model may be modeled in a separate, yet integrated decision model. Henceforth, we will refer to such data as "process-related data used by activities to make decisions". More specifically, we distinguish a set of BPMN process patterns that characterize process-related data used for making decisions in process models and suggest how such data can be represented in DMN decision models. Then, we provide a mapping of such BPMN patterns towards the corresponding elements of a DMN model. This mapping can be used by designers and analysts to (i) guide the extraction and separation of a decision model from a given process model and (ii) improve understanding of integrated process and decision models, always under a data perspective.

Our proposal deals with process modeling at conceptual level, aiming to improve communication between different stakeholders and to ease the refinement of existing process and decision models carried out by knowledge workers.

The steps that we followed to devise the proposed pattern-based approach can be summarized as follows. Firstly, (i) we identified a set of *decision patterns* that describe data elements commonly used in BPMN processes to represent data potentially used by activities to make decisions. To this end, we analyzed the representation and use of data in the BPMN and DMN standards, by also considering real-world processes. Then, (ii) we defined a formal mapping of the identified decision patterns towards dedicated (groups of) DMN elements. Finally, (iii) we discussed the post-processing of related decision and process models. To exemplify the applicability of our approach, we show the application of the discussed steps to a process taken from a real-world clinical domain [12].

Lying in the context of BPMN process modeling and analysis [13], the presented approach follows a high, conceptual level of abstraction, thus requiring expertise in domain knowledge to understand context-dependent process and decision modeling aspects. In this regard, we start from the assumption that knowledge about the "as-is" processes of an enterprise is available to decision analysts, regardless of whether it is gathered by multi-disciplinary teams or through direct interaction with stakeholders [5].

This paper is a comprehensive extension of previous work [14], which includes a detailed explanation of the proposed approach comprehensive of strengths and limitations, the complete analysis of the BPMN standard, and an extended application example.

The remainder of the paper is structured as follows. Section 2 discusses related work, focusing on the interplay between BPMN and DMN. Section 3 introduces foundational concepts by means of a clinical example. Section 4 describes the main steps of the proposed pattern-based approach. Section 5 explores how process-related data are represented in BPMN. Section 6 introduces the set of identified BPMN decision patterns. Section 7 presents the mapping of the distinguished BPMN patterns to DMN decision models. Section 8 discusses the application of our approach to a real healthcare process. Section 9 discusses strengths and limitations of the proposed approach. Finally, Section 10 concludes the paper.

2. Related Work

Despite having existed as an independent and evolving discipline [15], decision management is increasingly being used in conjunction with business process management to improve business outcomes and competitiveness. Organizations need to extract information and knowledge useful for decision-making and process improvement out of data being collected by their business processes and other (big) data sources [16, 17]. Besides, the increasing interest in knowledge-intensive processes, i.e., workflows whose conduct and execution are heavily dependent on knowledge workers performing interconnected decision-making tasks, demands proper support to ensure that the best decisions possible are made and that the collaboration between knowledge workers is fruitful [18].

However, since both decision and process management have long existed without proper integration, process modeling languages have been misused to encode decision logic [3, 5, 9]. As a result, existing process models often incorporate decisions that are encoded into control flow structures [5], hidden within process activities [3], or implicitly contained in process execution logs [10, 11]. This improper integration trend leads to maintainability, flexibility, re-usability, and scalability issues in both process and decision models [9, 19].

Therefore, separation of concerns has drawn increasing interest in the BPM field [2, 8, 15, 20], especially since the introduction of the DMN standard [6], designed to elicit and represent decision models that can possibly be used to complement BPMN process models, thus keeping concerns separate.

Several research approaches have addressed the design or derivation of DMN decision models that are complementary to BPMN process models with the aim of separating concerns, yet integrating the modeling of decisions and processes [1, 4, 5, 9, 12, 17, 21]. In [1], DMN is used in the context of collaborative networks to discern decisions that are incorporated in BPMN process models from those that can be modeled through more appropriate DMN diagrams. In [4], the authors propose a methodology for automatically deriving process models from Product Data

Models that capture the complex data dependencies underlying a workflow. The approach allows one to obtain a BPMN process model emphasizing the most important decisions, while detailed decision logic is outsourced to a dedicated DMN model. The extraction of decision logic from the process control flow is considered in [5]. The authors identify a set of control flow patterns often misused to capture decisions and show how DMN models succeed in reducing the complexity of a process model that embeds decision logic. In [17], the authors adopt event processing techniques to address the re-evaluation of decision based on updated and relevant real-time data represented by events, that may change the outcome of a decision being made during process execution. In [21, 12], the authors discuss how to jointly use the BPMN and DMN standards to represent the structured organizational aspects of real-world decision-intensive healthcare processes that include complex clinical and organizational decision-making.

Some of the introduced approaches focus on control flow aspects [1, 5]. Probably, this trend originates from the role of primary importance given in the BPMN standard to the so-called "control flow perspective". However, BPMN allows designers to represent data through data artifacts, events or text annotations [7, 22].

In this paper, we provide decision designers and analysts with an approach that eases the derivation of DMN models from data that are *explicitly* represented in BPMN process models and that provide input information for decision activities. In line with previous research [5], we assume that stakeholders have an active role in approving whether the identified decision activities are real business decisions.

However, dealing with data is more challenging for two main reasons. On the one hand, data contribute to build both explicit and implicit process knowledge [23] and, thus, they may contribute to decision-making in multiple ways beside being an input for decisions. On the other hand, the same data may be shared between process and decisions models, yet addressing different concerns. As a result, consistent process and decision models integration is of prime importance [9], as the same piece of information may be used in both models for different purposes and managed by different stakeholders. Lying at a conceptual level, our approach supports decision analysts and stakeholders in the identification of data relevant for decision making, but leaves the freedom to choose the most appropriate level of process and decision model integration.

The consistent integration of the BPMN and DMN standards has been the focus of some relevant contributions [3, 9, 19, 24]. In [3] the authors frame the role of a decision model within the context of a related business process and examine execution mechanisms for different availability of input data. Five novel principles for integrated process and decision modeling (5PDM) are proposed in [9] to guide designers in avoiding and solving inconsistencies between process and decision models, such as the unsound ordering of decision activities or the absence of input data. Consistency requirements between DMN decision models and BPMN process models are defined in [19]. The authors analyze multiple kinds of process models and, based on their structure and characteristics, provide a set of requirements for integrating DMN decisions. Finally, in [24] the authors identify common challenges related to the refactoring of process models that arise when integrating decision models.

Both the proposals presented in [19, 24] rely on the definition of relevant integration scenarios for process and decision models. In this paper, we take the view of having both decisions and processes intertwined within the same model, and start from the assumption of possibly having hidden decisions hard-coded in process models [5, 9, 24]. Despite considering process models having decisions as a local or global concern [19, 24], we focus on discovering decisions that have a process-restricted scope.

For example, let us consider the process of discharge planning from hospital following colorectal surgery. The information related to the patient's mobility and nutritional intake is used by care givers to plan how the patient should be treated in hospital and at home once discharged [25]. This information affects daily decisions related to transport arrangements (e.g., decide whether requiring night assistance for a patient that is not able to walk for more than 25 minutes unaided) and meal plans (e.g., decide whether precluding discharge if the patient does not tolerate solid meals), as well as higher-level managerial decisions related to re-admissions, long waiting lists, and bed-blocking (e.g., are there criteria related to patients' mobility that cause delayed discharge?).

In this paper, we focus on capturing and representing the first kind of exemplified decisions, that is, those made within a specific process and based on process-related data. These are sometimes referred to as local decisions [9]. The discovery of higher level, strategic decisions that are not explicitly made in business processes or span over multiple processes is out of the scope of this paper.

As argued in [9, 24], the main obstacle to consistent process and decision model integration appears to be the declarative nature of the DMN standard, which clushes with the dependency of process-related decisions from the invoking context. A solution to support separation of concerns while ensuring consistency between process and decision models is the representation of decisions as externalised services, following the principles of Service-Oriented Architecture (SOA) paradigm [26, 27]. According to this approach, business rules are grouped into a decision services that are incorporated within a web services layer to be consumed by the business process layer. For example, SOA^{+d} defines fine and coarse-grained services that are necessary for modeling the business, information system, and decisions of an organization [27].

More general approaches have investigated the integrated modeling and externalization of decision rules [28, 29]. In [28], the authors empirically evaluate a set of factors that affect the choice of whether business rules shall be incorporated into process models and propose a set of guidelines for improving the modeling of business rules. Similarly, in [29] the effects of business rule integration on business process model understanding is discussed and evaluated empirically.

In the field of decision mining, a framework to classify activities in a process model based on how the contribute to the overall decision dimension of a process is presented in [30]. The introduced approach enables an in-depth analysis of every activity in order to establish whether it entails a decision, and how it is related to other activities. Last but not least, other approaches presented in [10, 11, 31, 32] aim to semi-automatically extract complex decision logic from process event logs using decision trees and other conventional process mining techniques.

3. Motivating Example and Foundations

In this section, we introduce the BPMN [7] and DMN [6] standards through a suitable clinical example that summarizes the motivations behind our proposal. Then, we provide a formalization of the foundational concepts used in this paper.

Let us consider the process of diagnosing and treating patients affected by Chronic Obstructive Pulmonary Disease (COPD) [33], carried out by physicians and pulmonologists in a hospital setting. COPD is a chronic and irreversible condition of the lungs, caused by tobacco smoke or exposure to polluted environments. Hospital care for COPD is mostly focused on monitoring and reducing the patient's symptoms, whose severity determines which is the "stage" of the illness and, accordingly, how the patient must be treated.

In this paper, we consider dealing with the presentation in a hospital of patients complaining about respiratory discomfort suggestive of COPD. For simplicity, we consider patients that either need to be diagnosed with COPD, or that are known to have COPD and are experiencing a sudden worsening episode, i.e., a COPD exacerbation [21, 33]. The main steps of the introduced process are shown by the BPMN process model of Figure 1.

The process begins when start message event *Patient Request* Ⓐ is triggered upon receiving a patient request. In BPMN, start events initiate a process instance, while end events conclude it. Graphically, they are depicted as circles and may contain a marker to diversify the kind of trigger they react to (e.g, a message, a timer, a particular condition). For example, start message event *Patient Request*, depicted with an empty envelope marker, waits for the related message to be received to trigger the process.

Some events, such as message, signal, escalation, and error events, have the capability of carrying data. Among them, messages are used to depict the physical or information items exchanged during a communication between two participants [7]. In the studied setting, the request exchanged between a patient and a physician includes the patient's biographical data and the reason of presentation.

Process resource *Physician* Ⓑ, represented as a BPMN lane, conducts activity *Evaluate Request* by assessing the degree of emergency based on the patient request. In BPMN, activities represent work performed within the process. They are depicted as rectangles with rounded edges, and are distinguished in tasks, i.e., atomic activities, and subprocesses, that represent compound units of work. Activities may be decorated with markers that denote their type or loop characteristics.

In this paper, we focus on user activities, executed by a human performer, and business rule activities that interact with a business rule engine [7]. Graphically, user activities such as *Evaluate request* in Figure 1 include a human figure marker,

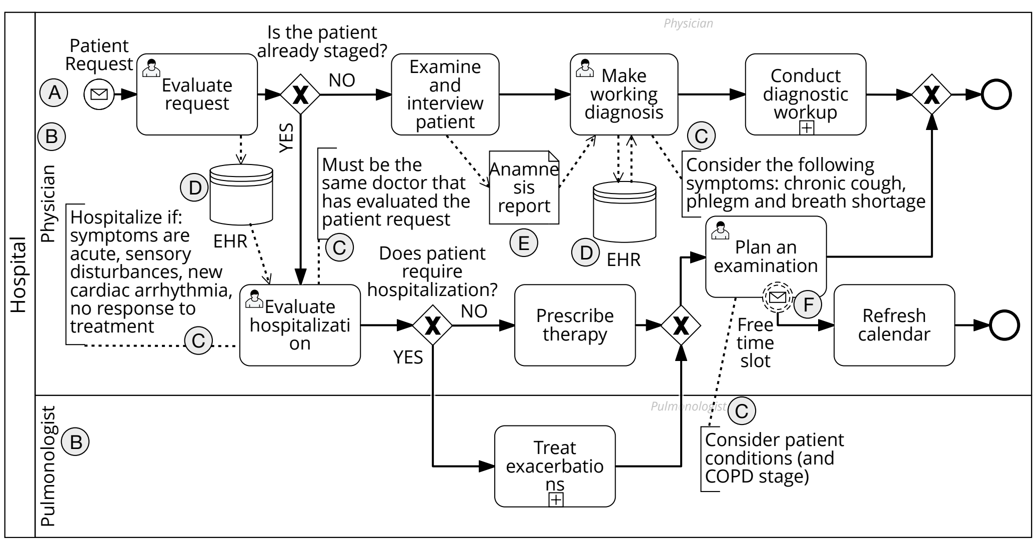

Figure 1: Sample BPMN process model for diagnosing patients with suspected Chronic Obstructive Pulmonary Disease, conducted in a hospital. Several kinds of data are used by process activities to make decisions: Ⓐ start message events; Ⓑ resources; Ⓒ text annotations; Ⓓ data stores; Ⓔ data objects; and Ⓕ boundary non-interrupting events.

while business rule activities have a marker that resembles a decision table.

The process flow is then split into two branches by an exclusive gateway, labeled by question *Is the patient already staged?*. In BPMN, gateways control the splitting and merging of the control flow and are depicted as diamonds having an internal symbol that denotes their routing behavior. Exclusive gateways are characterized by a "$\times$" symbol and represent a point in the process where one path is chosen among several alternative ones. Parallel gateways are characterized by a "$+$" symbol and are used to create or join parallel paths in the process.

If the patient is already staged, activity *Evaluate hospitalization* is conducted. Evaluating hospitalization is a decision that requires a physician to consider both the factors described in the associated text annotation Ⓒ and the patient history recorded in the Electronic Health Record *EHR* Ⓓ, represented as a BPMN data store. In hospitals, exacerbation treatment is under the care of a *pulmonologist*.

Otherwise, if hospitalization is not required, activity *Prescribe therapy* is conducted. Then, the physician must *Plan an Examination* to re-evaluate the patient. The date is chosen based on the patient conditions and on staff availability, recorded in the ward's calendar. If the notification of a new *Free time slot* Ⓕ appears during appointment scheduling, as depicted by the corresponding message boundary non-interrupting event, it is considered when deciding the day of the appointment. Boundary events are intermediate events attached to the border of an activity. They are represented as circles with a double border which is solid if the event is interrupting, i.e., its occurrence interrupts the on-going activity, or dashed if the event is non-interrupting, that is, a parallel exception path is enacted during activity execution [7].

Instead, if the patient is non-staged, the physician must *Examine and interview* the patient to collect data regarding symptoms, signs, and smoking habits. All these data are summarized in the *Anamnesis report* Ⓔ, depicted as a data object representing volatile data exchanged by process activities. These data are used by a physician to formulate a *Working Diagnosis*. Then, a diagnostic workup is conducted to either confirm a diagnosis of COPD or solve respiratory discomfort.

The BPMN process of Figure 1 shows how data represented by data objects, text annotations, data stores, and events are provided as an input for activities *Evaluate Request*, *Evaluate hospitalization*, *Make working diagnosis*, and *Plan an examination*. Since they all involve evaluation, planning, and other decision-intensive tasks such as clinical diagnosis, the mentioned activities are likely to use the associated data as an input for making decisions.

However, since process models are not meant to represent decisions, it is not always easy to understand whether such process-related data concern are also used

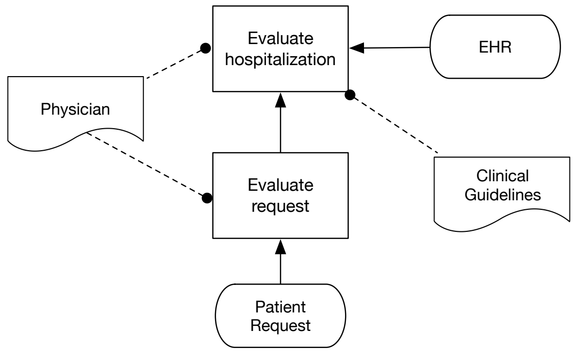

Figure 2: Example of DMN Decision Requirements Diagram, representing a decision *Evaluate hospitalization*, based on sub-decision *Evaluate request*, related input data and knowledge sources.

for making decisions and how.

Generally speaking, a decision is the act of determining an output value, from a number of input values, using decision logic to define how the output is determined from the inputs [6, 15].

In the DMN standard [6], decision models consist of two logical layers, one dealing with decision requirements, the other one with decision logic. On the one hand, decision requirements are modeled through a Decision Requirements Graph (DRG) that describes a domain of decision-making by specifying the network of decisions and their interdependencies. A DRG can be represented as one or more Decision Requirements Diagrams (DRDs) that may be used to present any particular view of the DRG. On the other hand, decision logic is used to describe in greater detail how decisions are made, potentially allowing the decision-making captured by decision requirements to be fully automated [6].

A DRD represents decisions, their interdependencies, and the data and knowledge on which they are based. As previously mentioned, a decision denotes the determination of an output from a number of inputs using decision logic [6].

Figure 2 shows an example of DRD related to the domain described by the process model of Figure 1. The two decisions *Evaluate request* and *Evaluate hospitalization* of Figure 2 are depicted as rectangles. A patient request is evaluated based on the patient's biographical data and on the reason of presentation.

If the patient is already staged and is experiencing an exacerbation, hospitalization must be evaluated following clinical guidelines to assess symptoms acuity and response to previous treatment. Accordingly, Figure 2 shows that the output of decision *Evaluate request* is used by decision *Evaluate hospitalization*, together with input data *EHR*, enclosing the results of the request evaluation and, if present, the patient stage. Input data denote the information used as an input by the decision and are depicted as a shape with two parallel straight sides and two semi-circular ends.

Finally, a knowledge source denotes an authority for a decision, which can either be a domain expert responsible for maintaining the decision (e.g., *Physician*) or source documents from which the decision is derived (e.g., *Clinical guidelines*) [6]. Graphically, knowledge sources are depicted as shapes with three straight sides and a wavy one.

The dependencies between DRD elements are expressed by different kinds of requirements. Information requirements connect input data (or decision outputs) with the decision that uses them and are depicted as solid arrows. Authority requirements denote the dependence of a DRD element on another DRD element that acts as a source of guidance or knowledge, and are depicted with a dashed arrow with a filled circular head. In Figure 2, decision *Evaluate hospitalization* is connected to knowledge source *Clinical guidelines* by an authority requirement.

The introduced example shows a DMN DRD that has been derived from a BPMN process model. In such a scenario, the connection between the process and decision models strongly relies on (1) decision activities and (2) data used within the process that also have potential decisional value. However, the identification of which process-related data may have decisional value and what concern (e.g., input data or knowledge source) they address when externalized in a dedicated decision model remains a challenging task for decision analysts and designers.

Since DRDs were devised to bridge business process models and decision logic, in this paper we focus on the decision requirements level and investigate how process-related data used for decision-making may be represented in DRDs.

In the remainder of this section, we introduce the formal definitions of process model and decision requirement diagram used in the rest of the paper.

Definition 3.1 (Process Model). A process model $m = (N, DN, C, TA, F, T, R,$ α_k, α_t, β, ρ, DA) is a tuple consisting of a finite non-empty set of control flow nodes N, a finite set of data nodes DN, a finite non-empty set C of control flow edges, a finite set of text annotations TA, a finite set of data associations F, a finite set of undirected associations T, and a finite set of resources R. The set $N = \{A \cup G \cup E\}$ of control flow nodes consists of the disjoint sets A of activities, G of gateways and $E = \{E_{start} \cup E_{int} \cup E_{end}\}$ of events. $E_{start} \subseteq E$ is the set of start events. $E_{end} \subseteq E$ is the set of end events. $E_{int} \subseteq E$ is the set of intermediate events that includes also the set of boundary events E_B, i.e., $E_{int} \supseteq E_B$. Control flow $C \subseteq N \times N$ connects the elements of N. $DN = DO \cup DS$ is the set of data nodes, consisting of the disjoint sets DO of data objects and DS of data stores. $F \subseteq (DN \times A) \cup (A \times DN)$ is the set of data associations that connect data nodes with activities. $T \subseteq (TA \times A) \cup (A \times TA)$ is the set of symmetric associations that connect activities with text annotations. α_k, and α_t are functions that associate a type to the elements of A. $\alpha_k : A \rightarrow \{task, subprocess\}$ distinguishes activities into tasks and subprocesses. Let us call $A' = \{a \,|\, a \in A$ and $\alpha_k(a) \mapsto task\}$ the set of tasks in m. Function $\alpha_t : A' \rightarrow \{abstract, user, business\, rule, service, script\}$ associates to each task a specific type. $\epsilon_k : E_B \rightarrow$ $\{$*interrupting*, *non-interrupting*$\}$ associates to each boundary event its interrupting behavior. $\beta : A \rightarrow 2^{E_B}$ is a function that associates to each activity $a \in A$ a set of boundary events attached to its border. Function $\rho : A \rightarrow R$ assigns to each activity the resource responsible for its execution. Finally, $DA \subseteq A$ is the set of decision activities such that $\forall da \in DA$ either $\alpha_k(da) = subprocess$ or, if $\alpha_k(da) = task$ then $\alpha_t(da) = user$ or $\alpha_t(da) = business\, rule$.

According to Definition 3.1 process models are static directed graphs with typed nodes. As for process execution, we refer to the token-based semantics and to the notion of activity life-cycle and operational semantics defined in BPMN [7].

In a nutshell, an activity is in state *ready* when a token arrives. Focusing on the happy path, when input dependencies are satisfied, the activity moves to state *active*. Then, once the work is complete and completing requirements are satisfied, the activity moves to state *completed*. For practicality, in this paper we refer to the time-span including state *active* and up to state *completed* as “running”. While an activity is *running*, boundary events may occur: if an interrupting event occurs, the activity moves to state *terminated*, whereas non-interrupting events do not directly affect the state of a running activity.

Definition 3.1 characterizes also decision activities. Decisions are often carried out in process tasks [6] or subprocesses [34], which we comprehensively refer to as “decision activities”. The DMN standard recommends that decision activities are executed either manually as in *user* tasks, or in a semi-automated manner, as in *business rule* tasks [6]. In BPMN, a *business rule* task provides a mechanism for the process to provide input to a business rule engine and to get the output of calculations that the business rule engine could provide. A *user* task is executed by a human performer. Accordingly, we consider both types of decision tasks, as reflected in Definition 3.1. Thereby, decisions may be represented as user or business rule tasks. Instead, decision subprocesses do not have a specified type, but rather enclose multiple steps of decision-making [34]. Further in this work, we omit the type of decision tasks when the context is clear.

Before moving on, it is worth noticing that not all user and business rule tasks represent a decision activity. Thus, although task types may help analysts in the identification of decision-making tasks, the approval of stakeholders is needed to properly identify decision activities [5].

As exemplified in the DMN standard [6] and in the motivating example of Figure 1, the name of the activity may suggest that decision-making is performed. For instance, when the label of a task starts with a verb that implies a decision, such as “decide”, “evaluate”, or “check”, then the activity is likely to be a decision activity. Also approaches coming from the world of decision mining and decision point analysis may support the identification of decision activities [32].

Our formalization of decision requirement diagram is presented below.

Definition 3.2 (Decision Requirement Diagram). A *Decision Requirement Diagram* (DRD) is a tuple (D, B, I, K, IR, KR, AR) consisting of:

- a finite non-empty set of *Decision* nodes D;
- a finite set of *Business knowledge* nodes B;
- a finite non-empty set of *Input data* nodes I;
- a finite set of *Knowledge source* nodes K;
- a finite non-empty set of directed edges IR representing *Information requirements* such that $IR \subseteq (I \cup D) \times D$;
- a finite set of directed edges KR representing *Knowledge requirements* such that $KR \subseteq B \times (D \cup B)$;
- a finite set of directed edges AR representing *Authority requirements* such that $AR \subseteq (D \cup I \cup K) \times (D \cup B \cup K)$.

Herewith, ($D \cup B \cup I \cup K, IR \cup KR \cup AR$) is a directed acyclic graph.

4. A Stepwise Pattern-based Approach

In this section, we describe the main design steps that led to the pattern-based approach proposed in this paper.

In BPM, the problem of having decisions hard-coded into process models gives rise to maintainability, flexibility, re-usability, and scalability issues [9, 19]. In particular, process models can contain a lot of data elements, such as data objects, data stores, or events, that may carry a hidden decisional value. Often, these data are connected to decision activities and used as an input for decision-making. However, by simply observing a process model it is difficult to understand which information shall be unbundled and included in a dedicated decision model.

As a possible solution to help decision analysts and designers to recognize these data elements and use them when designing a DMN decision model, we provide a set of BPMN patterns capturing the data perspective of process models and a mapping towards corresponding DMN DRD fragments.

To our knowledge, this work is the first one to focus on the data perspective of BPMN process models at a conceptual level, as other proposals have addressed the same problem dealing with control flow aspects [1, 5] or have focused on the separation of concerns by considering decision services [26, 27].

In this section, we describe the stepwise approach that we followed to design our solution, starting from a given BPMN process model and identifying the process-related data used by decision activities. By grounding our research on the established BPMN [7] and DMN [6] standards, while analyzing real-world scenarios, we provide an approach that is based on well-known design languages, while having empirical evidence in processes coming from different real-world healthcare domains [12, 35].

In the first step, we analyze the BPMN standard [7] and identify which process elements may carry data. The result of this analysis is a classification of elements with respect to their relevance in capturing data that can be used for decision-making. Then, in the second step, we define the decision patterns including the process elements discovered in the previous step. Finally, in the third step, we define the mapping of BPMN patterns towards possible DMN elements.

Step 1: Analysis of BPMN for Identifying Decision Patterns. For specifying a complete and well-grounded set of decision patterns, we conducted a systematic qualitative analysis of the BPMN standard [7] and identified which elements of the notation may carry data that are used by decision activities. In particular, we focused on visible elements and attributes, typically used in high-level process modeling. The detailed description of the BPMN standard analysis is reported in Section 5.

Step 2 Definition of Decision Patterns in BPMN. Starting from the results of Step 1, we defined and formalized a set of *decision patterns* capturing the data perspective of BPMN. Each pattern corresponds to a process fragment representing a decision activity based on process-related data, that can be extracted and represented in a separate decision model. In Section 6 we present the definition and formalization of the considered set of decision patterns.

Step 3: Mapping of BPMN Decision Patterns onto DMN DRDs. Once the main decision patterns are identified and defined, they shall be mapped to elements or fragments of DMN Decision Requirements Diagrams. In Section 7 we proposed and formalized the mapping between BPMN decision patterns and the corresponding elements of DMN DRDs.

The BPMN patterns classified in the second step can be used by analysts and designers to identify decisions in a process model, thus providing a baseline for process improvement. The mapping defined in the third step can guide the extraction of a set of DRD fragments from the process model, thus completing a first "as-is" analysis of the decision-making coordinated by a process [6].

Since such DRD fragments constitute an unadapted DMN decision model, designers must combine the obtained fragments by considering the correlation between different decisions dictated by the control and data flows of the process.

Then, the considered process model may be re-designed to make a more effective use of decision-making. During the creation of such a "to-be" specification, designers should evaluate whether process-related data must also remain in the process model or it is sufficient to keep them in the extracted decision model.

In Section 8, we discuss in detail the steps that should be carried out to use the proposed patterns to unbundle decisions from an existing process model.

5. Identification of Process-Related Data Used for Decision-Making

In this section, we discuss which kinds of data may be valuable for decision-making, starting from those explicitly represented in BPMN process models. Then, we complete our analysis by discussing the role of other kinds of process-related data for decision-making.

5.1. Qualitative analysis of the BPMN standard

In order to identify which process elements may carrying data that are used by decision activities in process models, we conducted a systematic qualitative analysis of the BPMN standard [7] and considered previous works addressing the suitability of BPMN for modeling decisions and (related) data [22, 36, 37].

The presented analysis aims to distinguish which BPMN elements capture data explicitly represented in process models, that can be used by decision activities for making decisions. We provide a selection of process elements relevant to modeling data valuable for decision-making, starting from data artifacts, events, and resources associated to decision activities.

Since we are considering decision activities as the main BPMN element directly or indirectly "connected" to data elements, our analysis has a defined scope.

Moreover, we consider only BPMN graphical elements that can be visualized in a process diagram, thus covering the whole BPMN Descriptive Process Modeling Conformance [7]. On the one hand, we have data flow elements (e.g., data objects, data stores) linked to decision activities. On the other hand, some control flow elements can be connected to decision activities and represent other kinds of data in a process model (e.g., events providing input data for decision activities).

In detail, we analyzed version v.2.0.2 of the BPMN standard [7] and considered 116 process elements in total.

The five basic categories of elements found in a BPMN process are: (1) Flow objects, (2) Data, (3) Swimlanes, and (4) Artifacts, and (5) Connecting objects. A short summary of the analyzed elements is presented in Table 1, where the relevance of BPMN elements for representing data that can be related with decision activities is denoted by a "+" symbol for full relevance, by a "–" symbol for full irrelevance, and by a "+/–" for partial relevance.

Whether process elements connected to decision activities really contain data useful for decision-making may depend on the modeled domain and on the specific process instance. However, at a conceptual level, we may assume that process designers and decision analysts have the domain knowledge expertise required to assess data relevance for decision-making.

We consider a group of process elements (e.g., activities, gateways) to be fully "relevant" with respect to the considered analysis goal, when all the objects belonging to that group may be connected to decision activities, thus potentially including data useful for decision-making. Similarly, we consider a group of process elements to be "partially relevant" with respect to the conducted analysis goals when (i) only part of the elements in that group satisfy our requirements (e.g., only catching events among all events) or (ii) the elements of the group are used to connect data-rich elements to decision activities (e.g., sequence flows connecting events to decision activities). Any group of elements that does not satisfy the above requirements is marked as "not relevant".

(1) *Flow objects* are the main graphical elements used to define the behavior of a business process. BPMN defines three kinds of flow objects:

- *Events.* Events represent facts that occur during process execution and affect the process flow. BPMN distinguishes events based on (a) their triggering behavior and (b) their position in the process. Some events such as message, escalation, error, signal, and multiple events have the capability to carry data, whereas others such as timer and conditional events may represent (temporal) conditions of the environment that may impact the way a decision is made.

Cat.	Elements	Relevance	Brief Summary
Flow Objects	Events	+/-	Only start events, intermediate catching events, and boundary events are relevant for our goal. If interrupting a boundary event is relevant when immediately followed by a decision activity located on its outgoing exception flow, if non-interrupting it is relevant when attached to a decision activity.
	Activities	-	Not relevant as activities are units of work: they may represent decisions, but not related data.
	Gateways	-	Not relevant as gateways are not meant to represent data. Exclusive gateways are points where a previously made decision is applied [5].
Data	Data objects Data inputs/outputs	+	Relevant. They must combined with data associations that connect them to decision activities.
	Data stores	+	Relevant. They must combined with data associations that connect them to decision activities.
Swimlanes	Pools	+/-	Relevant only if a pool consists of one lane because a decision activity can belong to only one lane.
	Lanes	+/-	Relevant when it contains data about roles responsible for executing and maintaining decisions.
Artifacts	Groups	-	Not relevant because groups represent only an informal visual mechanism for grouping elements, and they do not carry decisional data.
	Text annotations	+	Relevant. They must combined with data associations that connect them to decision activities.
Connecting objects	Sequence flows	+/-	Sequence flows do not represent data. However, when they are used to connect events to decision activities they are relevant for our analysis.
	Conditional flows	-	Not relevant, as they encompass automatic "routing decisions".
	Message flows	-	Not relevant. Message flows do not connect data to decision activities.
	Associations	+/-	Relevant, when combined with text annotations.
	Data associations	+/-	Relevant when used in combination with data objects, data inputs, data outputs, or data stores.
	Exception flow	+/-	Relevant only when it connects a boundary event with an immediately following decision activity.

Table 1: Relevance of BPMN elements for capturing data explicitly represented in process models that can be used by decision activities for making decisions. Full relevance is shown as a "+" symbol, partial relevance as a "+/–" symbol, and full irrelevance as a "–" symbol.

(a) Based on their triggering behavior, events are classified into *throwing* and *catching*, depending on whether they release (throw) a trigger or react to (catch) it. For catching events, output data become automatically available when the trigger occurs and can be used further in the process.

(b) Based on their position in the process flow, events are classified into *start*, *intermediate*, or *end* events. Intermediate events may also be attached to the boundary of activities and may have either an interrupting behavior or a non-interrupting one.

In this paper, we focus on events that may affect the execution of decision activities. Start events are relevant for our analysis since they are all catching events and, thus, they may carry data used by an immediately following decision activity, connected to the event by a sequence flow edge. Among intermediate events, we included only catching events, as throwing events are not suitable for our purpose. Intermediate events can be connected to a decision activity through a sequence flow (or an exception flow in case they attached to the boundary of a previous activity) or they can be attached to its boundary. In both cases, the decision activity may use the data carried by the event. However, in order for a decision activity to use data carried by a boundary event, the latter must be non-interrupting as an interrupted activity cannot use data. Since nothing can follow end events, they are not relevant for our purpose.

- *Activities.* An activity is used to represent work that is performed within the process. Since activities are not meant to represent data in processes, they are not relevant for our analysis.
- *Gateways.* Gateways are used to control the divergence and convergence of sequence flow in a process. Therefore, gateways are part of process control flow, so they do not represent data in processes. Even when exclusive gateways execute a data-based routing decision, the actual decision shall be made in the activity preceding the decision gateway [5] (i.e., gateways use the output of a previously made decision to route the flow and are sometimes referred to as "decision points").

(2) *Data* is represented by the following elements:

- *Data objects, data inputs, data outputs.* Data objects describe information that is needed for activities to be performed, or is produced during activity execution. Data inputs and data outputs provide the same information for processes. As we are considering data used by decision activities, data objects and data inputs connected to decision activities are naturally relevant for our selection and should be considered in combination with the data associations that link them. Instead, we exclude data outputs as they may represent the result of a decision, but not the used data. If the data output resulting from a decision is used as an input by another decision, it will be likely duplicated in the process or connected to both decision activities.

- *Data stores.* A data store allows activities to retrieve or update stored information that will persist beyond the scope of the process. Naturally, when connected to decision activities, data stores may provide information relevant for decision-making. Again, data stores should be linked to decision activities through data associations.

(3) *Swimlanes* provide a graphical account for participants in a process and group other flow objects in the following two ways.

- *Pools.* A pool is the graphical representation of participants of the process. It can consist of several lanes. A decision activity can belong to only one lane of the pool. The assignment of a resource to an activity is done by placing the activity within the selected lane of the pool.
- *Lanes.* Lanes are used to partition pools in order to organize and categorize activities. The assignment of a resource to an activity is done by placing an activity in the lane.

Despite they do not represent data directly, pools and lanes may be associated to decision activities. The information related to the (decisional) role assigned to the resource and corresponding data access restrictions may influence the outcome of a decision. Resources are relevant when they are also responsible for the governance of the decision-making.

(4) *Artifacts* are used to provide additional information about the process.

- *Groups.* BPMN defines groups as a "visual mechanism to group elements of a diagram informally" [7]. Groups cannot be connected to any BPMN elements, and they do not affect the process flow, but they rather serve to ease perception of process models to users. They are not relevant for our selection.
- *Text annotations.* Text annotations are a mechanism for a modeler to provide additional information in natural language to help the readers of a process model. When connected to decision activities, these artifacts can contain

significant information for decision-making (e.g., data, constraints on the way a decision activity is executed, or business rules [29]). Therefore, we include them into our selection, together with the associations connecting them to the corresponding decision activities.

(5) *Connecting objects* represent different ways of linking flow objects to each other or to additional information represented by artifacts. Although they do not represent data, they are needed to convey the information contained in events and to connect data artifacts to decision activities. Therefore, some connecting elements are given partial relevance.

- *Sequence flows.* A sequence flow can be used to show a partial ordering of activities in a process. As previously mentioned, we are interested in the sequence flows used to connect start/intermediate events that carry data to the decision activities that use them. For this reason, they are considered partially relevant.
- *Conditional flows.* A conditional flow is a special kind of sequence flow having a condition expression that is evaluated at runtime to determine whether that process path can be followed or not. Despite relying on the evaluation of a data-based condition, conditional flows encompass an automatic "routing decision". Since decision-wise their behavior is similar to that of exclusive gateways, they are not relevant for our selection.
- *Message flows.* A message flow is used to show the flow of messages between two participants that are prepared to send and receive them. BPMN defines a special kind of activities and events, called send and receive activities (events), to perform such information exchange. Since receive tasks in BPMN cannot represent decision-making, the only way to use the content of a message for decision-making is by considering catching message events properly connected to decision activities. Thus, the message flow is not relevant for our purpose.
- *Associations.* An association is used to link text annotations with other flow elements. When connected to a decision activity, text annotations can contain any kind of comment written in natural language. These comments may include data/information used by decision activities. Thus, we include them in our selection with partial relevance.
- *Data associations.* A data association is used to link data objects, data inputs, data outputs, or data stores with activities or events. Since data objects or data stores may contain data used by decision activities, the data associations connecting such data artifacts to decision activities are relevant for our purpose.

– *Exception flow.* An exception flow occurs outside the normal flow of the process and it originates from an intermediate event attached to the boundary of an activity that occurs during process execution. As we considered intermediate events followed by decision activities, we shall include the exception flow when it connects a boundary event with a following decision activity. For this reason, its relevance is partial.

Since we focus on high-level modeling, we considered elements and attributes that are visually represented in process models. Thus, we intentionally leave out non-visible attributes such as `InputSets` and `OutputSets` of activities [7], despite they represent process-related data.

5.2. Other Kinds of Data Used for Decision-Making

Here, we complete our analysis by considering (implicit) process-related data that are often used for decision-making. In detail, we discuss why some data (i) do not need to be externalized in decision models or (ii) are used for representing process-related information, without being explicitly included in BPMN models.

In general, not all kinds of data specified within process models need to be externalized in dedicated decision models. This happens when data elements include no decision input/knowledge or when execution/external events are used exclusively to automatically route the process flow. Indeed, decision models are often employed to represent and improve understanding of operational decisions, mostly focusing on decision algorithms and logics to support human decision-making. Therefore, lower-level process logic should not be represented by decision models at the requirements level.

As an example, let us consider catching events in the configuration of event-based gateways. In this case, event occurrence drives the process flow based on "instantaneous decisions" that are managed by process engines. Such kind of decisions made by a process engine should not be included in decision models, as they are based on process logic and routing rules, that depend on event processing.

For instance, let us consider an event-based gateway having events "Receive accept e-mail" and "Receive reject e-mail" in its configuration. Depending on whether the request is accepted or requested, the process behaves differently. Yet, the "decision" of which path must be taken is a mere reaction to a decision, made outside the scope of the process (in this case, made by the sender) and whose output events, determine how the process should proceed its execution.

From another perspective, information used for decision-making and included in decision models may not be explicitly represented in process models. For instance, domain knowledge, Key Performance Indicators (KPIs), or process execution logs often drive decision-making, but they are represented as meta-information or at a lower modeling level rather than being included in BPMN process models.

Reference data and domain knowledge. Human decision-making is performed by organization personnel, who interpret data according to their knowledge, personal experience, past organizational trends, and reference information enclosed within textual guidelines [12]. Indeed, background domain knowledge is often used to complement known decision inputs, before outputs are inferred [23].

Key performance indicators (KPIs). Indicators can be defined during process specification to evaluate the process execution performance and to measure the progress towards the achievement of business and organizational outcomes, based on specific objectives and milestones. KPIs can have a *local* or a *global* scope, depending on how and when they impact the process [21]. Local KPIs are often defined quantitatively to measure the process performance and can be used to adapt the process dynamically. An example of local KPI is activity duration: If a certain task lasts longer than expected, future process steps can be skipped. Conversely, global KPIs are defined qualitatively by aggregating information regarding several processes, and can be used to support higher level decisions, related to process re-engineering or role re-definition. For instance, process managers may decide to re-design part of a process, according to customers satisfaction reports. Decisions can be linked to the KPIs and objectives of the organization they impact. These, may coincide with the global KPIs referred by processes within which the decisions are made [38].

Execution information. Likewise KPIs, information extracted from process logs can also be used to dynamically adapt the process flow to prescribed execution requirements in order to achieve improved decision outcomes [32, 38].

Despite not being explicitly represented in BPMN process models, the information enclosed within domain knowledge, KPIs, and process logs may be integrated by decision designers once DMN models are created.

6. Decision Patterns Capturing the Data Perspective of BPMN

In this section, we define and formalize a set of patterns that combine the process elements selected during the analysis of BPMN with decision activities.

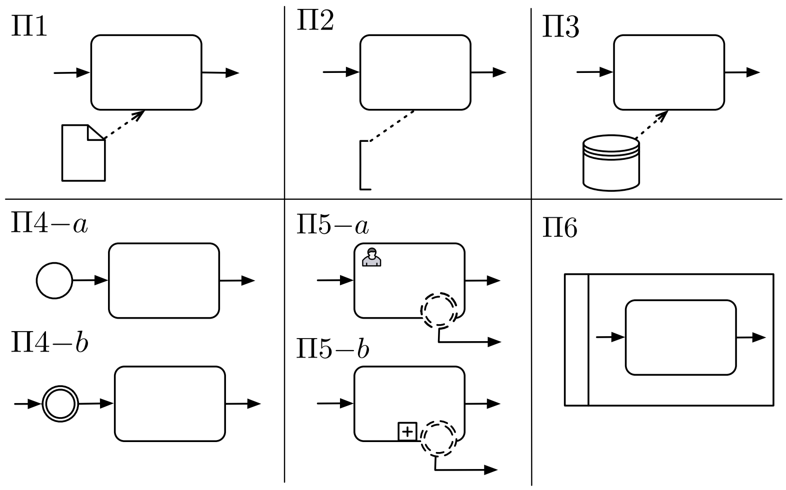

Figure 3: Summary of BPMN decision patterns. The possible plurality of the elements connected to a single decision activity is omitted for readability.

Figure 3 shows the set of decision patterns $\Pi 1 - \Pi 6$, that was derived from the systematic analysis of the BPMN standard [7] and whose empirical evidence was found in real-world healthcare process models from related research [5, 21, 35].

Each pattern corresponds to a process fragment, that is, a subset of the process model defined as a tuple. Thereby, each pattern contains a connected subgraph of a process model, which represents a decision based on process-related data that can be externalized into a separate decision model. The introduced patterns identify common ways to represent data related to decisions in process models. When the corresponding decision model is designed, a stakeholder (or a decision analyst having the required domain knowledge expertise) shall establish whether such data are significant for decision-making, as done for the patterns in [5].

For readability, we always show one process element of a kind attached to a decision activity, thus intentionally omitting the representation of multiple elements connected to a single decision activity. For example, pattern $\Pi 1$ shows only one data object connected with a decision activity, although the formalization of $\Pi 1$ provides that multiple data objects can be attached to a single decision activity. For each pattern, we provide its description, formalization, connection with the motivating example of Figure 1, and discuss possible variants.

It is worth noticing that we do not include decision outputs in our patterns. This gives us more flexibility when identifying decision activities as we select those that necessarily have some input data, but may or may not have an output explicitly represented in the process model. Indeed, decision outputs do not have to be explicitly represented as data elements in a process, as they can be captured by the output flows of an exclusive gateway [9]. Besides, the presence/absence of a decision output represented as data in the process does not change the structure of the obtained DMN decision requirements diagram.

Π*1* – Data objects used by a decision activity. Data objects represent data that is generated, consumed, and exchanged by process activities [7]. The BPMN standard does not say much about the inner structure of a data object. In practice, data objects can be used to capture a single data class of a databases, a collection of data classes, or a complex document [39]. Therefore, the information contained in data objects may or may not be used for decision-making and can be of any kind. When used for decision-making, the information contained in data objects is represented as input data for decision-making [5].

> Π*1* - DATA OBJECTS USED BY A DECISION ACTIVITY
> DESCRIPTION
> A decision activity uses the information contained within one or more data objects attached to it as input data for decision-making.
> ELEMENTS
> Decision activity da, data objects $do_1, \ldots, do_n$ attached to da through directed data associations $(do_1, da), \ldots, (do_n, da)$.
> FORMALIZATION
> Let $m = (N, DN, C, TA, F, T, R, \alpha_k, \alpha_t, \beta, \rho, DA)$ be a process model. A decision activity $da \in DA$ uses the set of data objects $DO' \subseteq DO$ if and only if $\forall\ do \in DO'$, $(do, da) \in F$. $\Pi 1$ is a process fragment that consists of decision activity $da \in DA$, a set of data objects $DO' \subseteq DO$, and a set of data associations $F_{DO'} = \{(do, da) \mid do \in DO'\} \subseteq F$.

As an example, let us consider an healthcare process, as the shown of Figure 1. A data object can represent the identifier of the patient, the list of previously prescribed therapies or a whole document, such as the anamnesis report. Regardless of data granularity, this information is related to a specific patient and it is represented as input data for the decision activity.

Π*2* – Text annotations used by a decision activity. Text annotations attached to activities are used to provide additional text information about that activity [7]. Therefore, as there is no limitation on their content, text annotations can represent both data, and requirements for decision-making. For instance, textual annotations may represent business rules integrated into process models [28].

According to our analysis, annotations are used in processes to capture comments related to which data can be used to execute the task, constraints on the execution of the task (such as deadlines, procedural aspects to be observed, or business rules), and information regarding activity actors and authorities for decision-making. Of course, the same text annotation can contain diverse kinds of data and not all of them may be needed for decision-making.

> **Π*2* - TEXT ANNOTATION USED BY A DECISION ACTIVITY**
> DESCRIPTION
> A decision activity uses the information contained within one or more text annotations attached to it as input data for decision-making or to provide details about decision sources or decision makers.
> ELEMENTS
> Decision activity da, text annotation $ta_1, \ldots, ta_n$ attached to da through undirected data associations $(ta_1, da), \ldots, (ta_n, da)$.
> FORMALIZATION
> Let $m = (N, DN, C, TA, F, T, R,$ α_k, α_t, β, ρ, DA) be a process model. A decision activity $da \in DA$ uses the set of text annotations $TA' \subseteq TA$ if and only if $\forall\ ta \in TA'$, $(ta, da) \in T$. $\Pi 2$ is a process fragment that consists of decision activity $da \in DA$, a set of text annotations $TA' \subseteq TA$, and a set of undirected associations $T' = \{(ta, da)|\ ta \in TA'\}$.

For example, let us consider the process of Figure 1. Text annotation "Consider patient conditions and stage", associated to decision activity *Plan an examination* suggests that the stage of COPD and the conditions of the patient must be considered during examination planning. In the considered domain, the more severe is the patient the earliest and most frequently medical checks should be planned. Instead, annotation *Must be the same doctor that has evaluated the patient request* does not provide any information useful for decision-making.

Π*3* – Data stores used by a decision activity. Data stores represent data that persist beyond the scope of the process [7]. Data stores are often used to represent databases that support process execution. Therefore, information retrieved from data store is likely to represent input data for decision activities.

In the process of Figure 1, data store *EHR* provides input data for decision activity *Evaluate hospitalization*. The same data store is connected also to decision activity *Make working diagnosis*.

Π*3* – DATA STORES USED BY A DECISION ACTIVITY
DESCRIPTION
A decision activity uses the information retrieved from one or more data stores as input data for decision-making.
ELEMENTS
Decision activity *da*, data stores $ds_1 \ldots ds_n$ attached to da through directed data association $(ds_1, da), \ldots, (ds_n, da)$.
FORMALIZATION
Let $m = (N, DN, C, TA, F, T, R, \alpha_k, \alpha_t, \beta, \rho, DA)$ be a process model. A decision activity $da \in DA$ uses the set of data stores $DS' \subseteq DS$ if and only if $\forall\ ds \in DS'$, $(ds, da) \in F$. $\Pi 3$ is a process fragment that consists of decision activity $da \in DA$, a set of data stores $DS' \subseteq DS$, and a set of data associations $F_{DS'} = \{(ds, da) \mid ds \in DS'\} \subseteq F$.

Π*4* – Event data used by a subsequent decision activity. In BPMN, some events such as message, escalation, error, and signal events have the capability to carry data [7]. In addition, timer events also encode temporal information that may have a value for business decision-making [40] and data obtained within a certain time frame may be used for re-evaluating decisions in real time [17].

When immediately followed by decision activities, the information carried by events can be used as input data for decision-making. According to the position of the triggered event, that is, either start or intermediate, we distinguish two variants of this pattern.

Π*4* – EVENT DATA USED BY A SUBSEQUENT DECISION ACTIVITY
DESCRIPTION
A decision activity uses the information carried by a previously occurred event as input data for decision-making.
VARIANTS

- $\Pi 4{-}a$ is a process fragment that consists of start event e, control flow $(e, da) \in C$, and decision activity da.
- $\Pi 4{-}b$ is a process fragment that consists of intermediate event e, control flow $(e, da) \in C$, and decision activity da.

FORMALIZATION

- $\Pi 4-a$: Let $m = (N, DN, C, TA, F, T, R, \alpha_k, \alpha_t, \beta, \rho, DA)$ be a process model. A decision activity $da \in DA$ uses data carried by a previously occurred start event $e \in E_{start}$ if and only if $(e, da) \in C$.
- $\Pi 4-b$: Let $m = (N, DN, C, TA, F, T, R, \alpha_k, \alpha_t, \beta, \rho,$ DA) be a process model. A decision activity $da \in DA$ uses data carried by a previously occurred intermediate event $e \in E_{int}$ if and only if $(e, da) \in C$.

For example, let us consider the process of Figure 1. The request of a patient is represented as a message start event that contains the patient's biographical data and the reason of presentation, and triggers the beginning process. In this case, the information included in the message is used by the subsequent decision activity *Evaluate request*.

Boundary interrupting events attached to an activity and leading to a decision activity located on the outgoing exception flow are a special case of pattern $\Pi 4-b$.

$\Pi 5$ – Boundary event data used by a decision activity. Sometimes boundary events can be directly attached to decision activities. If they interrupt the activity, the information that they carry cannot be used by the decision that has been interrupted. Instead, if the boundary event is non-interrupting the carried data may be used while the decision is being made: non-interrupting events trigger the corresponding event handlers that run in parallel with the on-going decision activity [40].

This consideration holds for user tasks or for subprocesses representing decisions and spanning for a certain amount of time, during which the boundary event can occur. On the contrary, a standalone business rule task representing a decision activity invokes the associated business rule or decision model upon activation [6, 7]. This is executed instantly, thus leaving no time for event occurrence.

For example, in Figure 1 non-interrupting boundary event *Free time slot* may notify that a new time slot is available in the agenda, while the physician is planning the future examination of a patient. This may occur whenever the hospital scheduling system detects that another patient has canceled or rescheduled an appointment. The physician may refresh the calendar and see whether the new availability may be an option for the considered patient: if the vacant time slot is too far away with respect to the severity of the patient, the physician may simply ignore this information.

Π5 – BOUNDARY EVENT DATA USED BY A DECISION ACTIVITY

DESCRIPTION

A decision activity uses the data carried by one or more non-interrupting boundary events as input data for decision-making.

VARIANTS

- $\Pi 5{-}a$ is a process fragment that consists of decision task da of type user and one or more non-interrupting boundary events $eb_1, \ldots, eb_n$ attached to its border.
- $\Pi 5{-}b$ is a process fragment that consists of decision subprocess da, and one or more non-interrupting boundary events $eb_1, \ldots, eb_n$ attached to its border.

FORMALIZATION

- $\Pi 5{-}a$: Let $m = (N, DN, C, TA, F, T, R, \alpha_k, \alpha_t, \beta, \rho, DA)$ be a process model. A running decision activity $da \in DA$ may be influenced by the occurrence of a set of non-interrupting boundary events $E'_B \subseteq E_B$ if and only if $\forall\, eb \in E'_B$, $\beta'(da) \mapsto E'_B$ where $\beta' : A \to 2^{E'_B}$ is the restriction of β to E'_B, eb occurs while da is running, $\epsilon_k(eb) \mapsto$ *non-interrupting*, $\alpha_k(da) \mapsto task$, and $\alpha_t(da) \mapsto user$.
- $\Pi 5{-}b$: Let $m = (N, DN, C, TA, F, T, R, \alpha_k, \alpha_t, \beta, \rho, DA)$ be a process model. A running decision activity $da \in DA$ may be influenced by the occurrence of a set of non-interrupting boundary events $E'_B \subseteq E_B$ if and only if $\forall\, eb \in E'_B$, $\beta'(da) \mapsto E'_B$ where $\beta' : A \to 2^{E'_B}$ is the restriction of β to E'_B, eb occurs while da is running, $\epsilon_k(eb) =$ *non-interrupting*, and $\alpha_k(da) = subprocess$.

Π6 – Decision activity associated to a specific role/resource. In BPMN, lanes are used to partition and organize the activities of a process and, often, they are used to represent resources, internal roles, systems or departments [7].

Here, we take the view that the information regarding internal roles associated to a decision activity may be used to determine who is the authority for the decision and which must be his or her expertise. Accordingly, the DMN standard defines knowledge sources "to model governance of decision-making by people (e.g., a manager), regulatory bodies (e.g., an ombudsman), documents (e.g., a policy booklet) or bodies of legislation (e.g., a government statute)" [6].

Besides, it is likely that different process roles may have diverse decisional power, also based on access privileges to sensitive information. If the same decision can be made by resources having different/hierarchical roles in the process, these may correspond to multiple/hierarchical authorities in a decision model.

The process of Figure 1 is contained in one main pool *Hospital* partitioned into two lanes, *Physician* and *Pulmonologist*. Whereas pulmonologists are not responsible for the governance of decision-making, for crucial decision activities, such as diagnosis, *Physicians* have a dual role. In the process they are responsible for taking care of the patient and for making the right clinical decisions, while in the decision model, they are responsible for defining the treatment and diagnostic steps, and for maintaining clinical guidelines.

> **Π*6*** – DECISION ACTIVITY ASSOCIATED TO A SPECIFIC RESOURCE
> DESCRIPTION
> A decision activity is executed by a resource having a specific process role. The information related to the role can be used to determine whether the decision maker is also an authority for the decision.
> ELEMENTS
> Decision activity da and associated resource r.
> FORMALIZATION
> Let $m = (N, DN, C, TA, F, T, R, \alpha_k, \alpha_t, \beta, \rho, DA)$ be a process model. A decision activity $da \in DA$ is executed by a process resource $r \in R$ having a specific role if and only if $\rho(da) = r$.

7. Mapping BPMN patterns to DMN Decision Requirements Diagrams

In this section, we introduce the formal mapping between the set of proposed decision patterns and DMN decision requirements diagrams.

To this end, we define the set $\Delta = \{\Delta 1, \ldots, \Delta 6\}$ of DRD fragments, which corresponds to the set of decision patterns $\Pi = \{\Pi 1, \ldots, \Pi 6\}$.

Our mapping is based on a correspondence relation $\Gamma = \{\Gamma 1, \ldots, \Gamma 6\}$, such that $\Gamma \subset \Pi \times \Delta$. The DRD fragments $\Delta 1, \ldots, \Delta 6$ are subgraphs contained in a DRD as a tuple, such that $d \in D$, $I' \subseteq I$, $K' \subseteq K$, $IR' \subseteq IR$, $AR' \subseteq AR$.

The correspondence relation Γ is visualized with the help of correspondence graphs in Figure 4 and a detailed discussion of the mapping is provided below. Since some BPMN elements may be mapped to multiple DMN elements, a few DRD elements are considered to be optional, meaning that, even if they have been identified as data useful for decision-making, not all the corresponding BPMN elements have to always be mapped to them. Figure 4 depicts such DRD elements with a gray-shaded filling. For readability reasons, we do not show the possible plurality of elements of the same kind connected to a decision activity.

In detail, the correspondence relation Γ always maps decision activity $da \in DA$ of each BPMN fragment constituting a decision pattern to decision $d \in D$ of the corresponding DRD fragment. Bearing this in mind, below we discuss only the correspondences of the other elements for each mapping.

Γ*1* A mapping $\Gamma\mathit{1}$ is a correspondence relation between the BPMN pattern $\Pi\mathit{1} = (da, DO', F_{DO'})$ and the DRD fragment $\Delta\mathit{1} = (d, I', IR')$. Each data object $do \in DO'$ is mapped onto input data $i \in I'$ since they both represent operational data used by a decision (activity). Each corresponding data association $(do, da) \in F_{DO'}$ corresponds to information requirement $ir \in IR'$.

Γ*2* A mapping $\Gamma\mathit{2}$ is a correspondence relation between the BPMN pattern $\Pi\mathit{2} = (da, TA', T')$ and the DRD fragment $\Delta\mathit{1} = (d, I', K', IR', AR')$. Text annotation $ta \in TA'$ corresponds to input data $i \in I'$ whenever it represents operational data needed for decision-making. In this case, undirected association $t \in T'$ is mapped to information requirement $ir \in IR'$. Alternatively, a text annotation may include information related to documents used for making the decision. In this case, $ta \in TA'$ is mapped onto knowledge source $k \in K'$ whenever it represents a non-functional requirement for decision-making. In this latter case, undirected association $t \in T'$ is mapped onto authority requirement $ar \in AR'$. Yet, since text annotations do not always represent both input data and knowledge sources, in the DRD fragment i and k are represented as optional, as highlighted by the shading.

Γ*3* A mapping $\Gamma\mathit{3}$ is a correspondence relation between the BPMN pattern $\Pi\mathit{3} = (da, DS', F_{DS'})$ and the DRD fragment $\Delta\mathit{3} = (d, I', IR')$. Each data store $ds \in DS'$ corresponds to input data $i \in I'$ as data stores are used to represent databases used by decision activities. Each data association $(ds, da) \in F_{DS'}$ corresponds to information requirement $ir \in IR'$.

Γ*4* A mapping $\Gamma\mathit{4}$ is a correspondence relation between the BPMN pattern $\Pi\mathit{4} = (da, e, (e, da))$ and the DRD fragment $\Delta\mathit{4} = (d, i, ir)$. The mapping considers both pattern variants $\Pi\mathit{4}-a$ and $\Pi\mathit{4}-b$, as they have the same formal structure. Event e is either a start or an intermediate event that carries data influencing the decision, and it corresponds to input data $i \in I$. The

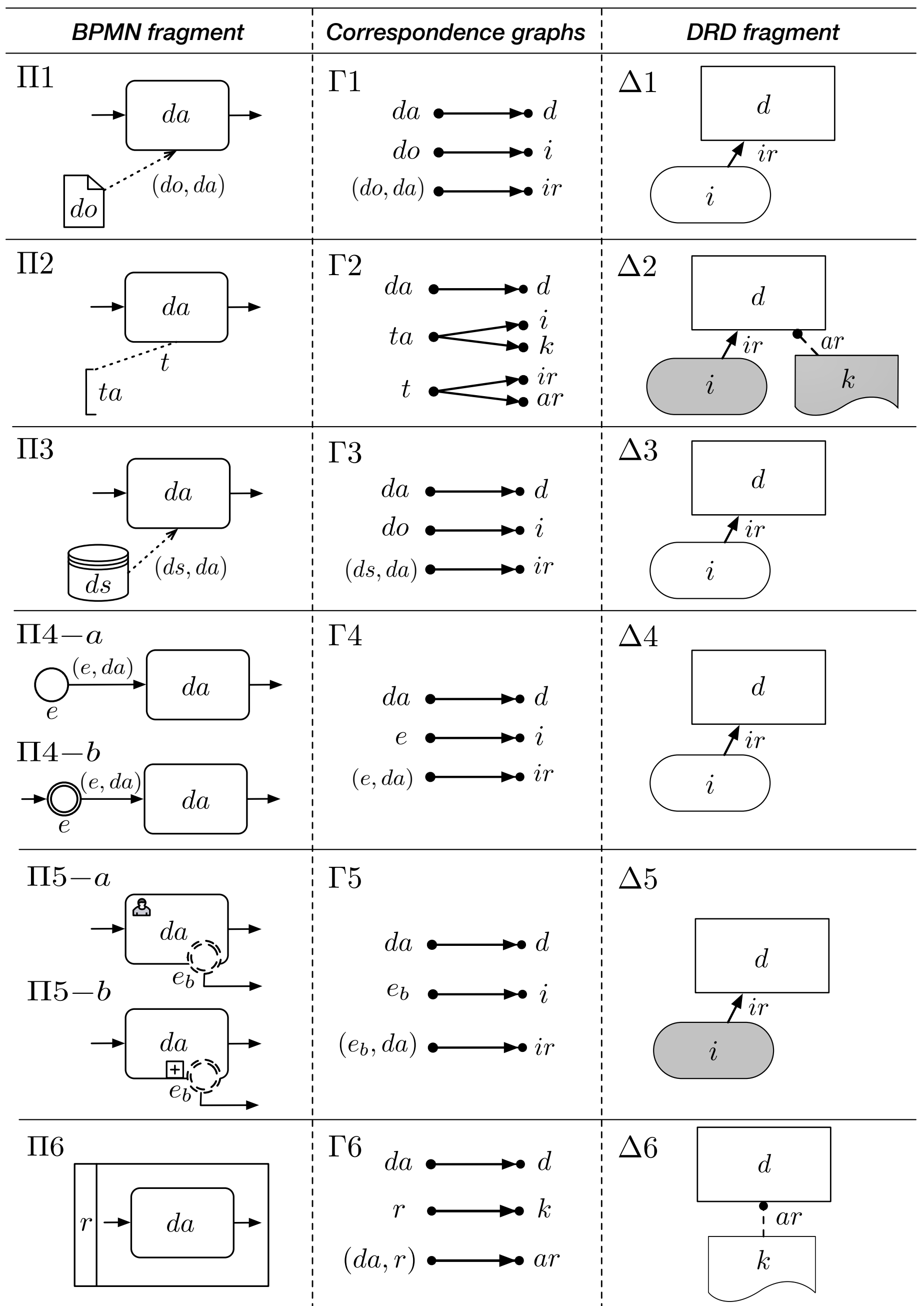

Figure 4: Mapping of the introduced BPMN patterns to corresponding DRD fragments. The shading of the DRD shapes means that the elements are optional for modeling and execution.

corresponding control flow edge $(e, da) \in C$ is mapped onto information requirement $ir \in IR$.

Γ5 A mapping $\Gamma 5$ is a correspondence relation between the BPMN pattern $\Pi 5 = (da, E'_B, \beta')$ and the DRD fragment $\Delta 5 = (d, I', IR')$. Each boundary event $e_b \in E'_B$ carries data used for making the decision, and it corresponds to input data $i \in I'$. Similarly, the corresponding relation $\beta'(da) \mapsto E'_B$ which associates boundary events to one activity is mapped onto information requirement $ir \in IR'$, even if it is not visualized in the process diagram. Since in both cases the boundary event may not occur, the corresponding input data element i in the DRD fragment is shown as optional.

Γ6 A mapping $\Gamma 6$ is a correspondence relation between the BPMN pattern $\Pi 6 = (da, r, (da, r))$ and the DRD fragment $\Delta 6 = (d, k, ar)$. Resource r is mapped to knowledge source $k \in K$ if it represents a non-functional requirement for decision-making (e.g., the authority responsible for decision governance), and then $\rho(da) \mapsto r$ should be mapped onto authority requirement $ar \in AR$.

The choice of representing resources as knowledge sources comes from the definition of the latter ones in the DMN standard [6]. Knowledge sources may be domain experts responsible for defining or maintaining them, source documents from which business knowledge models are derived, or sets of test cases with which the decisions must be consistent.

In addition, all *decisions* belonging to the introduced DRD fragments can also reference *business knowledge models* enclosing decision logic [6]. This is recommended if the decision logic is reused by multiple decisions. In this case, the corresponding *knowledge requirements* should also be provided. However, since business logic is typically not present in graphical process models, we did not include business knowledge models in our mapping.

8. Applying the Pattern-based Approach to Existing Process Models

In this section, we describe the different steps that decision analysts may follow when applying the introduced pattern-based approach to derive a DMN decision model related to an existing BPMN process model.

For each step, we discuss the design challenges and limitations, and show its application to a real-world example taken from the clinical domain of Chronic Obstructive Pulmonary Disease (COPD) [33].

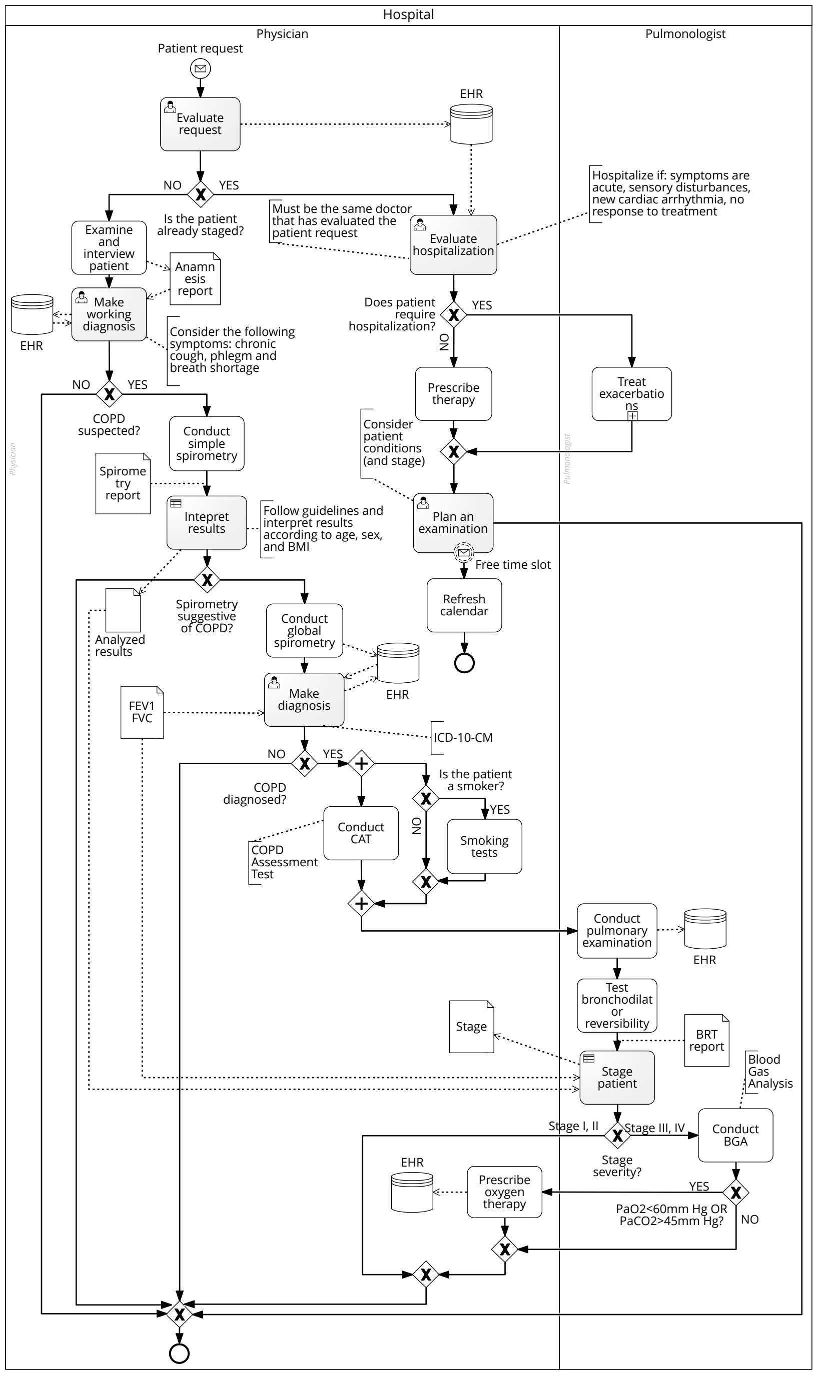

Figure 5: Process for dealing with unplanned presentation of patients possibly affected by Chronic Obstructive Pulmonary Disease (COPD). Decision activities are shaded.

8.1. The Studied Healthcare Domain

As partly introduced in Section 3, COPD is an irreversible condition of the lung requiring continuous multidisciplinary care. In the remainder, we focus on the BPMN process depicted in Figure 5, corresponding to the one of Figure 1 where the previously collapsed subprocess *Diagnostic workup* is expanded. The depicted process model focuses on the unplanned presentation of patients in a hospital complaining of acute respiratory symptoms that can be referable to COPD.

The process has been modeled by interviewing the personnel of a German hospital, and by combining the retrieved information with clinical practice guidelines [33] and with results obtained by previous research [21, 12]. Interviews were conducted by adopting the approach introduced in [12] and knowledge about clinical practice was acquired during process design by directly observing how physicians operate in hospital. In the presented case study, one of the main goals of process design was to support the identification and standardization of critical steps in the treatment of COPD and of the local and global decisions underlying the whole care process [12]. Thus, decisions were also identified by designers during process design and were submitted to stakeholders for approval.

In detail, we asked stakeholders targeted questions, regarding which and how many decisions could be discerned in the processes, whether and how these would influence the process flow, and which were the data requirements for decision-making. In the studied context, hospital physicians play the most important role in the process in terms of decision-making, as the outcome of their decisions is often used as an indicator of process quality [12]. In addition, they are also responsible for maintaining the decisions. For specialized pulmonary assessment and for the related decision-making physicians are helped by pulmonologists.

The process of Figure 5 begins when a *Patient request* is received by a physician, who determines whether the patient has already been diagnosed with a certain stage of COPD or a new diagnosis for the present condition is required. This decision is portrayed by user task *Evaluate request*, and it is mostly based on the information included in the request and on the physical conditions of the patient.

If the patient is experiencing an exacerbation causing symptom worsening, hospitalization is evaluated, based on the patient's history recorded in the Electronic Health Record (EHR) and on current symptoms. If the patient does not require hospitalization, therapy is prescribed, otherwise the worsening is treated in hospital. In either case, a future examination is planned: the chosen date depends on the patient's conditions and on doctors' availability, which is constantly updated as explained in Section 3.

If the patient has not been previously diagnosed with COPD, respiratory symptoms and risk factors must be collected through an *Examination with Interview* that serves as a basis for medical diagnosis. A diagnosis of COPD consists of two phases: (i) a working diagnosis, which requires the physician to identify signs that may lead to COPD suspicion based on the collected anamnesis data, and (ii) a definitive diagnosis of COPD, which may confirm the working diagnosis based on the results several medical tests.

Spirometry is required by international clinical guidelines to make a diagnosis of COPD. Simple spirometry provides information about the post-bronchodilator forced expiratory volume in one second (FEV1) and the forced vital capacity (FVC) of the patient. When the ratio between these two values is lower than 0.7, the presence of airflow limitation that is not fully reversible is confirmed. Spirometric results are interpreted according to predefined thresholds, but they also have to take the patient age, sex, and Body Mass Index (BMI) into account [33].

Global spirometry provides more accurate and detailed results, and it is used to assess COPD severity. Spirometric outcome interpretation is a delicate decision as it is the main piece of information used by physicians to *Make diagnosis* and by pulmonologists to *Stage patient*.

Once COPD has been diagnosed, additional tests and examinations, such as the COPD Assessment Test (CAT), smoking tests, and a pulmonary examination are conducted to grade the severity of the disease according to specific stages, defined in clinical guidelines [33]. COPD staging is represented as a business rule task, conducted under the supervision of a pulmonologist, who classifies the patient into one of the four admissible stages based on the results of the previously conducted spirometry and of a bronchodilator reversibility test.

Finally, after staging the patient, arterial blood gas analysis is conducted to see if temporary or permanent oxygen therapy must be administered.

In the following subsections, we refer to this example and explain how to (i) identify the decision patterns described in Section 6 in a process model, (ii) apply the mapping described in Section 7 to derive the corresponding DRD fragments, (iii) obtain the final DMN model and, where appropriate, consider process model refactoring or decision model post-processing.

8.2. Identification of Decision Patterns in a Process Model

First of all, a decision analyst should detect all the decision activities in a process, if this was not already done during process design. To this end, the analyst may rely on his or her domain knowledge, or may consult stakeholders.

This step can be supported by the natural language analysis of activity labels [41] that recommends candidate decision activities in a process model. Also decision points used for decision mining typically correspond to decision activities [9, 32].

Another way to facilitate this first identification step is to detect the exclusive split gateways in a given process model and consider the activities immediately preceding the gateways as candidate decision activities. Again, a stakeholder should confirm that the selected activities represent real business decisions [5]. Then, for each identified decision activity, the analyst should check whether it is connected to any events, data nodes, text annotation, or resource in accordance with the patterns $\Pi 1$–$\Pi 6$ described in Section 6.

Example. The analysis of the BPMN process model of Figure 5 was conducted with the help of stakeholders and led to the detection of seven decision activities, listed in Table 2.

Then, we extracted the process fragments corresponding to the introduced BPMN decision patterns, as shown in the upper part of the four rows of Figure 6.

8.3. Mapping of BPMN patterns to DRD fragments

Finding patterns $\Pi 1$–$\Pi 6$ in a process model does not always imply that the discovered data are valuable for decision-making. Indeed, patterns capture the representation of data in a process model, but do not provide details about the value of such data. Thus a decision analyst should always refer to the application domain to assess whether the identified pieces of information are actually involved in the decision-making. This also depends on the level of detail of the considered process model: if a process model is designed to include a lot of technical information, some data connected to decision activities may not be an input for decisions, as they may be needed in the process for other purposes such as, for instance, conformance checking or activity execution.

In particular, special care is needed for text annotations ($\Pi 2$), boundary non-interrupting events ($\Pi 5$), and resources ($\Pi 6$).

Text annotations are often informal comments, written in natural language, and no limit is set on their content. For this reason, they may contain useful information as well as additional data that is not useful for making decisions. Domain knowledge expertise, often resulting from the interaction with stakeholders, is necessary to discern useful data in this context.

For example, in the process of Figure 5 decision activity *Evaluate hospitalization* is linked to two different text annotations, one "Hospitalize if: symptoms are acute, sensory disturbances, new cardiac arrhythmia, no response to treatment"

Decision activity	Identification of decision patterns
Evaluate request	It uses the information related to the patient and cause of presentation, carried by start message event *Patient Request*, as input data ($\Pi 4-a$). The physician who evaluates the incoming patient is responsible for making and maintaining the decision ($\Pi 6$).
Make working diagnosis	It is mostly based on data regarding the patient's health condition, gathered from the *Anamnesis report* ($\Pi 1$), represented as a data object, and from the *EHR* ($\Pi 3$). All data are interpreted according to clinical guidelines, summarized by the linked text annotation ($\Pi 2$). A physician is the authority responsible for the working diagnosis ($\Pi 6$).
Interpret results	It takes data object *Spirometry report* as an input ($\Pi 1$) and evaluates its content based on clinical guidelines ($\Pi 2$), but also taking into consideration the patient's age, sex, and BMI ($\Pi 2$). This latter information is used as input data and, thus, it is represented as a DRD input in Figure 6, while guidelines are mapped onto a knowledge source.
Make diagnosis	It relies on the results of global spirometry and on the working diagnosis, both stored in the EHR ($\Pi 3$). The results of global spirometry are compared with reference values FEV1 and FVC, represented as a data object ($\Pi 1$). Text annotation *ICD-10-CM* denotes diagnosis encoding, thus being irrelevant decision-making. A physician is responsible for making the diagnosis and is also an authority for that decision ($\Pi 6$).
Stage patient	Staging is carried out by a pulmonologist, who classifies the patient into a specific COPD stage, depending on the previously made spirometry interpretation results, on the and on the information contained in the *BRT report*, both represented as data objects ($\Pi 1$). Process resource *Pulmonologist* is not responsible for the governance of patient staging and, thus, it is excluded from the selection of the patterns.
Evaluate hospitalization	It must consider the patient's current and past symptoms, and the history of previous exacerbations or treatments, as stated in the attached text annotation "Hospitalize if: symptoms [...]" ($\Pi 2$). Data regarding both previous exacerbations and symptoms are retrieved from the EHR ($\Pi 3$), where the output of activity *Evaluate request* is also recorded. Physicians are responsible for evaluating hospitalization and for maintaining that decision ($\Pi 6$). Text annotation "Must be the same doctor that has evaluated the patient request" does not contain information valuable for decision-making and, thus, it is discarded.
Plan an Examination	It is executed by a physician ($\Pi 6$), who considers the patient's stage as an input, as described in the attached text annotation ($\Pi 2$). Besides, real-time data about physicians availability must be considered: if a *free time slot* becomes available it is used during the planning ($\Pi 5-a$).

Table 2: Decision activities identified in the process of Figure 5 and related decision patterns.

provides useful information for deciding whether the patient should be hospitalized, the other one "Must be the same doctor that has evaluated the patient request" contains information that is needed only for constraining activity execution.

The same considerations may be done for boundary non-interrupting events, as the received trigger may or may not contain data that have the potential to change the decision outcome at run-time.

Being an intrinsic part of process models, resources are usually not considered in decision models. However, since process resources may have a decisional role, they should also be represented in decision models, especially when they are also responsible for the governance of a decision. Particularly, if the process resource is also domain experts responsible for defining or maintaining the decision he or she also makes, then the BPMN resource should be mapped towards a DMN knowledge source.

Example. In the example of Figure 5, physicians have a dual role: in the process they are responsible for taking care of the patient and for making the right clinical decisions, while in the decision model, they are responsible for defining the treatment and diagnostic steps, and for maintaining clinical guidelines.

For the considered scenario, all of the process resources detected by pattern $\Pi 6$ (i.e., physicians) were mapped to knowledge sources, as process resources where directly responsible of decision governance. Pattern $\Pi 2$ is mapped onto knowledge sources or input data, depending on the kind of information it includes. Pattern variants $\Pi 4-b$ and $\Pi 5-b$ are not found, as the considered process does not have intermediate events nor decision subprocesses. The obtained DRD fragments are shown in the lower part of the four rows of Figure 6.

8.4. Post-Processing of Decision Models and Adaptation of Process Models

Once a set of the DRD fragments is derived, two additional post-processing steps need to be carried to obtain a decision model that represents business decisions previously encoded in the process.

In the first place, a complete decision requirement graph needs to be constructed by combining the obtained fragments into one or more DRDs. This should be done by taking into account the inter-dependencies between different decisions, some of which are dictated by the process control flow or by the use of shared output/input data, and by eliminating repeated elements.

For example, when a decision activity produces data objects as an output that are reused as inputs by another decision activity, an information requirement should be added between the two decisions [5]. Similarly, decision activities may

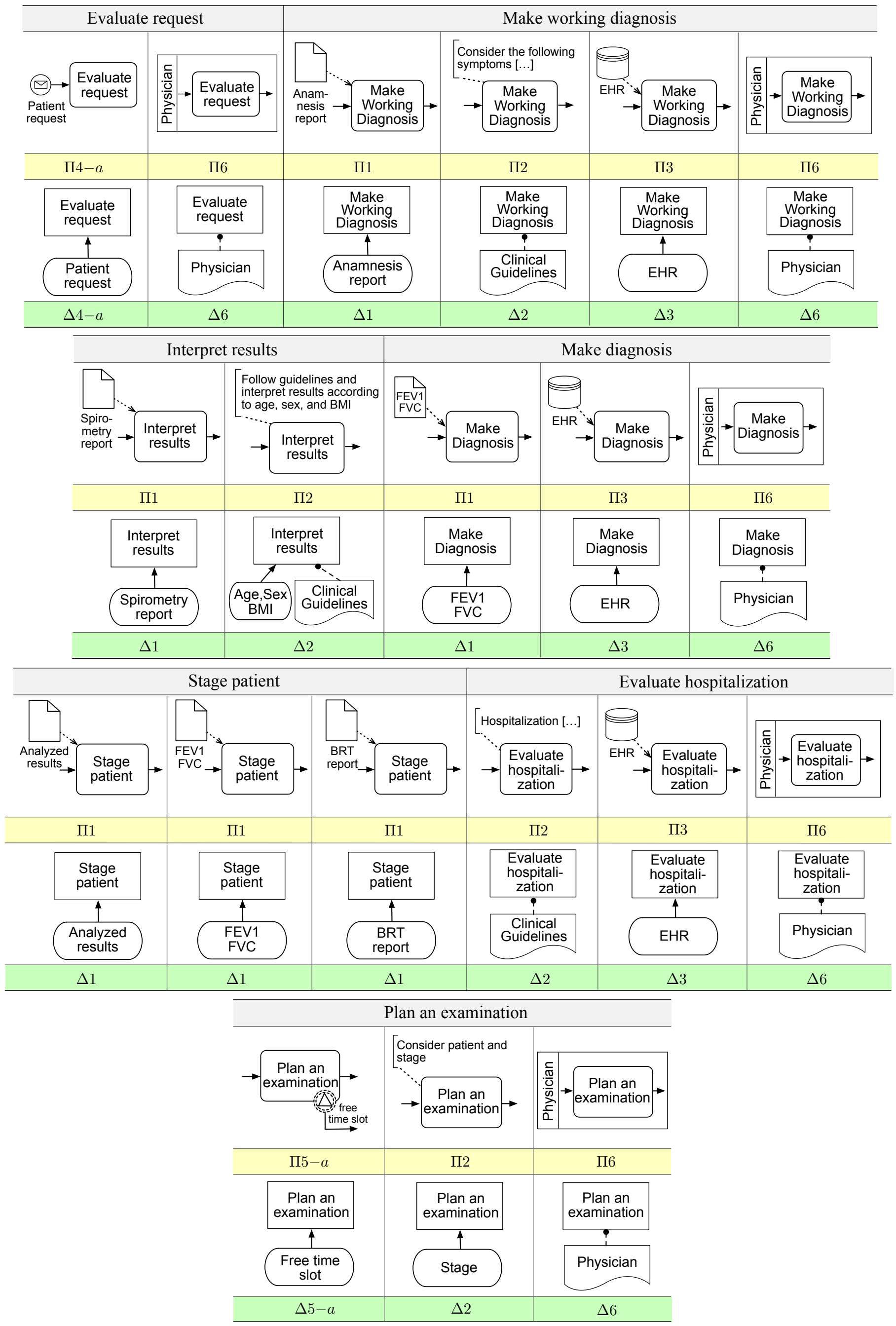

Figure 6: Decision patterns extracted from the decision activities listed in Table 2 and corresponding DRD fragments, obtained by applying the mapping explained in Section 7.

be related by reading and writing operations on a data store. If no connection with other decisions can be determined, the DRD fragment shall remain independent.

In some cases, multiple combinations of the extracted DRD fragments are suitable to describe one scenario and it is up to decision designers to choose the more appropriate one, by following stakeholders indications.

Once the comprehensive DRD is constructed for a given decision model, the original process model may be adapted to improve the representation of data and decision activities. Usually process adaptation is carried out to reduce inconsistencies and improve the integration of process and decision models [9].

In the considered context, process adaptation involves mostly data representation. On the one hand, if some data are used exclusively for decision-making and are already captured by the newly designed DRD, an analyst may decide to remove them from the process to increase readability. On the other hand, missing data used as an input for decision-making shall be added to the process to ensure correct process and decision enactment [9]. For instance, the output of a decision activity a_1 representing an intermediate result needed by decision activity a_2 shall be represented as a data object in the process model or, at least, a_1 and a_2 should have access to a shared data store.

Compared to the misuse of control flow for representing decision-logic [5, 9], the removal of data from the process model is not always mandatory when dealing with data, as process-related data do not directly influence process execution and the same piece of information may represent different concerns in different models. For example, it is absolutely acceptable to have a data object representing both an input for a decision activity in a process model, and an input to a decision in a DRD [6]. Indeed, if not all required data are included in the process model, the decision activities requiring that data will not be executed properly [9].

Of course, designers may consult stakeholders to determine whether it is relevant to keep the data element that carries a decisional value in the process model (e.g., for documentation purposes) or eliminate it from the process model (e.g., if a text annotation is misused to describe a decision rule that is has become a business rule or a knowledge source in the decision model obtained after the mapping). Lastly, process resources being also authorities for decisions should never be removed from process models, as they have a dual role, which is correctly captured by the integrated process and decision models.

Eventually, for the determined decision activities of the process model, undirected association links to the corresponding elements of the extracted DRD model should be added. Once decisions have been extracted from the process model, a decision engine can be employed to evaluate the decisions at runtime. Actually,

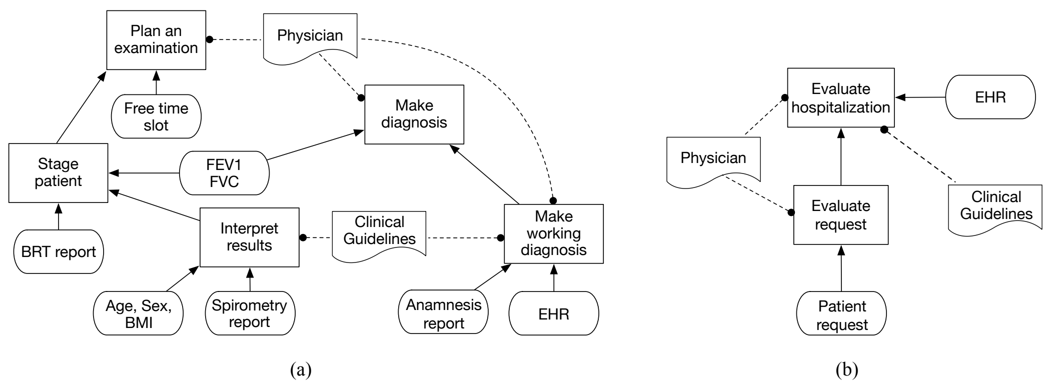

Figure 7: DRDs obtained by composing the DMN fragments depicted in Figure 6.

we do not require a dedicated process engine, but could rely on an arbitrary implementation of the process.

Example. The DRD fragments discovered from the process of Figure 5 were consequently combined together and compiled into two separate decision requirements diagrams, presented in Figure 7. In order to link different decisions, we considered both the relationships between decision activities dictated by the process control flow [5] and the flow of information (i.e., input/output relations and access to shared data) that links such activities.

As an example, let us consider Figure 7(a). Decision activity *Make working diagnosis* produces as an output a preliminary diagnosis based on data stored in the patient's EHR and enclosed in the anmnesis report, which are interpreted according to clinical guidelines. Such preliminary diagnosis is combined with the results of simple spirometry, analyzed during decision activity *Interpret Results*, and global spirometry, both stored in the EHR and used as input for decision activity *Make diagnosis*. This flow of information is mostly realized through data store *EHR*, but also the process control flow sets a partial order between activities *Make working diagnosis* and *Make diagnosis*. Thereby, one information requirement is added to the DRD of Figure 7(a) connecting decision *Make working diagnosis* to decision *Make diagnosis*.

Once the patient is diagnosed, he or she needs to be staged, based on the results of the bronchodilator reversibility test (BRT) and spirometric assessment. Therefore, activity *Stage patient* takes the output of activity *Interpret results* (i.e., data object Analyzed results) as an input, together with the BRT report. In Figure 7(a) this output/input relation is shown by the added information requirement between decisions *Interpret results* and *Stage patient*. Finally, decision activity *Plan an ex-*

amination determines when the patient must be seen again by the physician, based on the assessed stage of the diseases and availability. Thus, activity *Plan an examination* uses the output of activity *Stage patient* as an input. In Figure 7(a) this connection is shown by means of information requirement that goes from decision *Stage patient* to decision *Plan an examination*.

Instead, Figure 7(b) shows an information requirement between decisions *Evaluate request* and *Evaluate hospitalization*, together constituting an independent DRD. These two decisions are dependent from one another since the patient request includes information regarding the patient, such as the history of previous exacerbations, directly used to evaluate the need for hospitalization. In addition, activity *Evaluate request* is directly connected to activity *Evaluate hospitalization* through a decision structure represented by exclusive gateway *Is the patient already staged?* immediately following decision activity *Evaluate request*.

The extracted DMN model in Figure 7 serves as an explanatory decision model for the BPMN process model of Figure 5, as it explicitly incorporates the process-related data used by decision activities for decision-making. Herewith, the extracted decision model can be executed complementarily to the process model, and thus, the principle of separation of concerns is observed. However, the original BPMN process model should contain the undirected association links between decision activities and the corresponding elements of the extracted DMN decision model, a step that can be done at the implementation level.

8.5. Empirical Findings

To gather empirical evidence supporting the reliability and applicability of the proposed pattern-based approach, we analyzed the relationship between process-related data and decisions in selected real-world process models.

We manually analyzed a repository of 43 process models addressing a complex procedure of liver transplantation [35]. This set of process models resulted from a collaborative design effort, involving both practitioners and process modeling experts, aimed to enable process monitoring and analysis.

The goal of our analysis was to quantify how often in practice the patterns described in Section 6 are used for capturing or supporting decision-making.

All patterns $\Pi 1 - \Pi 6$ were detected in the 43 process models, except for $\Pi 3$. The absence of pattern $\Pi 3$ can be explained by the fact that the domain experts involved in process design were hospital physicians who did not directly interact with IT-systems. Moreover, although a consistent amount of information was recorded in the hospital IT system, the connection between process and data management systems was not made explicit.

The most commonly detected patterns were $\Pi 2$, which was found in 44.19% of the process models, and $\Pi 1$, which was present in 39.53% of them. These results met our expectations, as the use of text annotations ($\Pi 2$) and data objects ($\Pi 1$) to describe input data for decision activities is quite common in healthcare processes. Indeed, clinical decision are often based on medical knowledge and evidence, stored in patients health records and interpreted according to professionals' experience and expertise [12]. Having multiple and fragmented information sources to consider makes clinical decisions hard to be represented in process models. As a result, text annotations are often used to make BPMN processes easier to be understood by practitioners, who are used to read and interpret textual documents such as clinical guidelines.

Patterns $\Pi 5{-}a$ and $\Pi 5{-}b$ occurred quite frequently in the analyzed process models. In detail, $\Pi 5{-}a$ was detected in 25.58% of the repository process models, while $\Pi 5{-}b$ was found in 18.60% of them. However, during our analysis, we observed that boundary events were overused for modeling non-exceptional control flows, probably instead of exclusive gateways. Therefore, the probability of finding $\Pi 5{-}b$ in other application domains may be lower.

Patterns $\Pi 4{-}a$, detected in 11.63% of the process models, and $\Pi 4{-}b$, found in 16.28% of them, were slightly less common. This can be explained due to the fact that start and intermediate events preceding decision activities, usually serve as triggers for decisions rather than bearing decision-related data.

Pattern $\Pi 6$ representing the involvement of a process resource as an authority for decision-making was detected in 11.63% of the process models. This percentage was lower than we expected. However, we discovered that in the studied context there was no differentiation among the resources entitled to execute each clinical pathway (i.e., process) and to make the related decisions. That is, since the correspondence resource-pathway was almost one-to-one, resources were not explicitly modeled as BPMN swimlanes and this explains why we could not detect $\Pi 6$ as often as expected.

After detecting the decision patterns, we applied the steps described in Sections 8.2 $-$ 8.4 to obtain the decision models related to the 43 processes. During this phase, we followed the principles outlined in [9] to gain a deeper understanding of decision model post-processing and process model re-design.

Overall, we discovered that several decisions were hidden in the analyzed process models. This fact may be explained by at least two reasons.

On the one hand, the process models were designed under time pressure in a series of workshops [35] and, thus, multiple aspects were incorporated within one (process) model. On the other hand, designers were not aware of the principle of separation of concerns, especially because process models were designed before the introduction of the DMN standard [6].

In addition, most of the detected decisions were based on process-related data originating from manual activities. Such data had been annotated on the process models to improve understanding and to serve as a basic reference for compliance analysis. As a result, the data-centric decision patterns occurred quite frequently in the considered repository and most of the identified process fragments were suitable to be externalized into dedicated decision models.

Overall, analyzing real-world process models allowed us to shed light on the features of process models that are a prerequisite for successfully applying the proposed pattern-based approach. We discuss them in Section 9, together with the strengths and limitations of our work.

9. Discussion and Limitations

The pattern-based approach proposed in this paper is grounded on the consolidated BPMN and DMN standards [6, 7] and partly complements previous research addressing the discovery of decisions from the process control-flow [5]. Indeed, by considering the data perspective of BPMN process models from a decision modeling standpoint, our approach contributes to providing analysts with complete overview of the decision-making carried out in a process.

The choice of relying on design patterns is convenient for two main reasons. Firstly, BPM researchers and practitioners are familiar with the use of patterns to support process technology [5, 22, 42]. Secondly, experts have noticed that there are often recognizable patterns in the decisions they make. This is particularly true for clinical and healthcare domains, where decision-making is often standardized for improving compliance with care protocols [12, 25, 35, 43].

In our case, patterns are useful to provide a conceptual basis of process and decision design. The decision patterns described in Section 6 capture high-level concepts that are commonly understood in business environments and, thus, they may also serve to ease communication among designers, analysts, and practitioners, and to encourage collaborative process and decision (re-)design. Last but not least, the proposed approach allows designers to focus on the extracted decisions that can be evaluated at run-time by a decision engine.

Being based on the data perspective of BPMN process models, our approach depends on the quality of the process models used as a starting point.

In particular, we assume that (i) process-related data have been modeled *explicitly* (i.e., by using process elements that can be visualized in a process model) and consistently, and that (ii) the decision activities based on such data are somehow distinguishable in the process model. Our first assumption stems from the fact that the use of data objects or text annotations in BPMN processes is considered good practice to improve process model understanding [9, 44, 45] and, thus, process models typically include them. We intentionally leave out non-visible elements and attributes, e.g., `InputSets` and `OutputSets` of activities [7].

The two assumptions mentioned above go hand in hand with the assumption that domain knowledge is available to designers and analysts. Typically, in real-world settings, domain experts have deep understanding of the parts of the process they perform, whereas designers often lack of domain knowledge [44].

However, the gathering of domain knowledge is usually carried out during process discovery [12, 44] and, thus, the knowledge related to existing process models is likely to be already available when our approach is applied. Besides, process and decision analysts may actively interact with stakeholders to gain a clear understanding of the application domain, identify decision activities, and ensure that process-related data are properly and explicitly modeled.

Since the scope of our patterns is centred around decision activities, which typically precede local "decision points" in process models [5], we intentionally leave out decisions spanning multiple processes. This point of view may be seen as a limitation of our approach, as the scope of the discovered decisions is restricted to the considered process model and, particularly, to its data perspective.

Moreover, when combining the obtained DRD fragments into a complete graph, intermediate decision results that were not visualized in the starting process model are likely to be missing in the decision model.

To be able to add them while maintaining process and decision model consistency, we must include all decision activity outputs in the process model. Forcing such outputs to be explicitly represented in process models may as well be perceived as a limitation. Yet, according to the principles for consistent process and decision model integration outlined in [9], all intermediate results necessary for correct process and decision enactment shall be represented in the process model.

All things considered, the proposed pattern-based approach considers a high level of abstraction. We focus on modeling the decision requirements derived from process models to identify the extent of the decision-making and the authorities involved in it. Then, we discuss the re-design of the process to ensure that the decision-making it coordinates is consistently specified.

The full specification of the decision logic is not aimed at. Consequently, we do not address the automation and implementation of the models into software components. For the same reason, we do not consider the decision-service layer, although it seems promising to automate decisions while keeping process and decision logic separate [26].

In light of the assumptions discussed above, the empirical findings presented in Section 8.5 have some generalization limitations.

Since the process models were designed for monitoring and analysis purposes, special attention was given to identify the data sources involved in the process at design time. Thus, most of the 43 processes we analyzed were rich in explicitly represented data. In addition, we had easy access to the domain knowledge needed to identify decision activities, thanks to the complete documentation that was gathered during process design, the availability of clinical guidelines, and the direct interaction with practitioners.

Both these aspects, which are prerequisites to apply the proposed pattern-based approach, are not generalizable to every process repository.

10. Conclusion

In this paper, we proposed a pattern-based approach to support process and decision analysts in unbundling decisions hidden in BPMN process models, focusing on the data perspective of the latter ones.

In a nutshell, we distinguished a set of decision patterns that characterize process-related data used for making decisions in existing process models.

The patterns were elicited by conducting a systematic qualitative analysis of the BPMN standard [7] and by considering previous work on the suitability of BPMN for modeling decisions and (related) data [22, 36, 37]. Then, we provided a mapping of such patterns onto corresponding DRD fragments. Finally, we discussed the derivation of a comprehensive DRG from the obtained fragments and considered process model re-design to make an effective use of the data needed for decision-making and to ensure process and decision model consistency [9].

Previous research considered the extraction of decisions from the process control flow [1, 5] or focused on the decision services layer to separate process and decision logics [26, 27]. By focusing on the extraction of DMN models starting from the data perspective of BPMN process models, the presented approach contributes to enrich the stream of proposals aimed to improve the separated yet integrated use of the BPMN and DMN standards.

For future work, we plan to extend the mapping proposed in Section 7 in order to consider additional information about decision-making in processes (e.g., business knowledge models) and to provide contact points with existing approaches that consider also decision logic [5, 26]. Besides, we aim to tackle the conceptual representation of a wider range of process-related data used for decision-making, such as domain knowledge and KPIs.

References

[1] T. Biard, A. Le Mauff, M. Bigand, J.-P. Bourey, Separation of decision modeling from business process modeling using new Decision Model and Notation (DMN) for automating operational decision-making, in: Working Conference on Virtual Enterprises, Springer, 2015, pp. 489–496. doi:10.1007/978-3-319-24141-8_45.

[2] J. Taylor, J. Purchase, Real-World Decision Modeling with DMN, Meghan-Kiffer Press, 2016.

[3] L. Janssens, J. De Smedt, J. Vanthienen, Modeling and enacting enterprise decisions, in: International Conference on Advanced Information Systems Engineering, Springer, 2016, pp. 169–180. doi:10.1007/978-3-319-39564-7_17.

[4] H. van der Aa, H. Leopold, K. Batoulis, M. Weske, H. A. Reijers, Integrated process and decision modeling for data-driven processes, in: Business Process Management Workshops, Vol. 256 of Lecture Notes in Business Information Processing, Springer International Publishing, 2016, pp. 405–417. doi:10.1007/978-3-319-42887-1_33.

[5] K. Batoulis, A. Meyer, E. Bazhenova, G. Decker, M. Weske, Extracting decision logic from process models, in: Proceedings of the 27th International Conference on Advanced Information Systems Engineering (CAiSE), Vol. 9097 of Lecture Notes in Computer Science, Springer, 2015, pp. 349–366. doi:10.1007/978-3-319-19069-3_22.

[6] Object Management Group, Decision Model And Notation (DMN), v. 1.1, Available at: http://www.omg.org/spec/DMN/1.1/ (2016).

[7] Object Management Group, Business Process Model and Notation (BPMN), v. 2.0.2, Available at: http://www.omg.org/spec/BPMN/2.0.2/ (2013).

[8] D. L. Parnas, On the criteria to be used in decomposing systems into modules, Communications of the ACM 15 (12) (1972) 1053–1058. doi:10.1145/361598.361623.

[9] F. Hasic, J. D. Smedt, J. Vanthienen, Augmenting processes with decision intelligence: Principles for integrated modelling, Decision Support Systems 107 (2018) 1–12. doi:https://doi.org/10.1016/j.dss.2017.12.008.

[10] E. Bazhenova, S. Buelow, M. Weske, Discovering decision models from event logs, in: International Conference on Business Information Systems (BIS), Vol. 255 of Lecture Notes in Business Information Processing, Springer, 2016, pp. 237–251. doi:10.1007/978-3-319-39426-8_19.

[11] M. De Leoni, M. Dumas, L. García-Bañuelos, Discovering branching conditions from business process execution logs, in: Fundamental Approaches to Software Engineering (FASE), Vol. 7793 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 114–129. doi:10.1007/978-3-642-37057-1_9.

[12] C. Combi, B. Oliboni, A. Zardini, F. Zerbato, A Methodological Framework for the Integrated Design of Decision-Intensive Care Pathways—an Application to the Management of COPD Patients, Journal of Healthcare Informatics Research 1 (2) (2017) 157–217. doi:10.1007/s41666-017-0007-4.

[13] M. Weske, Business Process Management - Concepts, Languages, Architectures, 2nd Edition, Springer, 2012.

[14] E. Bazhenova, F. Zerbato, M. Weske, Data-centric extraction of DMN decision models from processes, in: Business Process Management Workshops, Vol. 308 of Lecture Notes in Business Information Processing, Springer International Publishing, 2018, pp. 542—-555. doi:10.1007/978-3-319-74030-0_43.

[15] B. Von Halle, L. Goldberg, The decision model, Taylor and Francis Group, 2010.

[16] S. F. Wamba, D. Mishra, Big data integration with business processes: a literature review, Business Process Management Journal 23 (3) (2017) 477–492. doi:10.1108/BPMJ-02-2017-0047.

[17] L. Pufahl, S. Mandal, K. Batoulis, M. Weske, Re-evaluation of decisions based on events, in: Enterprise, Business-Process and Information Systems Modeling, Springer International Publishing, Cham, 2017, pp. 68–84. doi:10.1007/978-3-319-59466-8_5.

[18] C. Di Ciccio, A. Marrella, A. Russo, Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches, Journal on Data Semantics 4 (1) (2015) 29–57. doi:10.1007/s13740-014-0038-4.

[19] L. Janssens, E. Bazhenova, J. De Smedt, J. Vanthienen, M. Denecker, Consistent integration of decision (DMN) and process (BPMN) models, in: Proceedings of the CAiSE '16 Forum, Vol. 1603 of CEUR Workshop Proceedings, CEUR-WS.org, 2016, pp. 121–128.

[20] T. Debevoise, J. Taylor, J. Sinur, R. Geneva, The micro-guide to process modeling and decision in BPMN/DMN, CreateSpace Independent Publishing Platform, 2014.

[21] C. Combi, B. Oliboni, A. Zardini, F. Zerbato, Seamless design of decision-intensive care pathways, in: IEEE International Conference on Healthcare Informatics (ICHI), IEEE, 2016, pp. 35–45. doi:10.1109/ICHI.2016.9.

[22] P. Wohed, W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede, N. Russell, On the Suitability of BPMN for Business Process Modelling, in: Business Process Management, Vol. 4102 of Lecture Notes in Computer Science, Springer, 2006, pp. 161–176. doi:10.1007/11841760_12.

[23] D. Calvanese, M. Dumas, F. M. Maggi, M. Montali, Semantic DMN: formalizing decision models with domain knowledge, in: Proceedings of the International Joint Conference on Rules and Reasoning (RuleML+RR), 2017, pp. 70–86. doi:10.1007/978-3-319-61252-2_6.

[24] F. Hasić, L. Devadder, M. Dochez, J. Hanot, J. De Smedt, J. Vanthienen, Challenges in refactoring processes to include decision modelling, in: Business Process Management Workshops, Springer International Publishing, Cham, 2018, pp. 529–541. doi:10.1007/978-3-319-74030-0_42.

[25] J. F. Fiore, A. Bialocerkowski, L. Browning, I. G. Faragher, L. Denehy, Criteria to determine readiness for hospital discharge following colorectal surgery: an international consensus using the delphi technique, Diseases of the Colon & Rectum 55 (4) (2012) 416–423. doi:10.1097/DCR.0b013e318244a8f2.

[26] A. Zarghami, B. Sapkota, M. Z. Eslami, M. van Sinderen, Decision as a service: Separating decision-making from application process logic, in: 2012 IEEE 16th International Enterprise Distributed Object Computing Conference, 2012, pp. 103–112. doi:10.1109/EDOC.2012.21.

[27] F. Boumahdi, R. Chalal, A. Guendouz, K. Gasmia, Soa^{+d}: a new way to design the decision in soa—based on the new standard decision model and notation (dmn), Service Oriented Computing and Applications 10 (1) (2016) 35–53. doi:10.1007/s11761-014-0162-x.

[28] W. Wang, M. Indulska, S. Sadiq, To integrate or not to integrate – the business rules question, in: Proceedings of the 28th International Conference on Advanced Information Systems Engineering (CAiSE), Vol. 9694 of Lecture Notes in Computer Science, Springer International Publishing, Cham, Switzerland, 2016, pp. 51–66. doi:10.1007/978-3-319-39696-5_4.

[29] W. Wang, M. Indulska, S. Sadiq, B. Weber, Effect of linked rules on business process model understanding, in: Proceedings of the Business Process Management: 15th International Conference (BPM 2017), Vol. 10445 of Lecture Notes in Computer Science, Springer International Publishing, 2017, pp. 200–215. doi:10.1007/978-3-319-65000-5_12.

[30] J. De Smedt, F. Hasić, S. K. L. M. vanden Broucke, J. Vanthienen, Towards a holistic discovery of decisions in process-aware information systems, in: Proceedings of the 15th International Conference on Business Process Management (BPM 2017), Vol. 10445 of Lecture Notes in Computer Science, Springer International Publishing, 2017, pp. 183–199. doi:10.1007/978-3-319-65000-5_11.

[31] F. Mannhardt, M. De Leoni, H. A. Reijers, W. M. Van der Aalst, Decision mining revisited - discovering overlapping rules, in: Proceedings of the 28th International Conference on Advanced Information Systems Engineering (CAiSE), Vol. 9694 of Lecture Notes in Computer Science, Springer, 2016, pp. 377–392. doi:10.1007/978-3-319-39696-5_23.

[32] A. Rozinat, W. van der Aalst, Decision mining in ProM, in: Business Process Management, 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006, Proceedings, Vol. 4102 of Lecture Notes in Computer Science, 2006, pp. 420–425. doi:10.1007/11841760_33.

[33] B. R. Celli, W. MacNee, A. Agusti, A. Anzueto, B. Berg, A. S. Buist, P. M. Calverley, N. Chavannes, T. Dillard, B. Fahy, et al., Standards for the diagnosis and treatment of patients with COPD: a summary of the ATS/ERS position paper, European Respiratory Journal 23 (6) (2004) 932–946. doi:10.1183/09031936.04.00014304.

[34] E. Bazhenova, M. Weske, A data-centric approach for business process improvement based on decision theory, in: Proceedings of the 15th International Conference on Business Process Modeling, Development and Support, Vol. 175, Springer, 2014, pp. 242–256. doi:10.1007/978-3-662-43745-2_17.

[35] K. Kirchner, N. Herzberg, A. Rogge-Solti, M. Weske, Embedding conformance checking in a process intelligence system in hospital environments, in: Process Support and Knowledge Representation in Health Care, Vol. 7738 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 126–139. doi:10.1007/978-3-642-36438-9_9.

[36] M. Chinosi, A. Trombetta, BPMN: An introduction to the standard, Computer Standards and Interfaces 34 (1) (2012) 124–134.

[37] M. Zur Muehlen, J. Recker, How much language is enough? Theoretical and practical use of the business process modeling notation, in: Seminal Contributions to Information Systems Engineering, Vol. 5074 of Lecture Notes in Computer Science, Springer, 2013, pp. 429–443. doi:10.1007/978-3-540-69534-9_35.

[38] E. Bazhenova, M. Weske, Deriving decision models from process models through enhanced decision mining, in: Business Process Management Workshops, Vol. 256 of Lecture Notes in Business Information Processing, Springer, 2015, pp. 444–457. doi:10.1007/978-3-319-42887-1_36.

[39] A. Meyer, M. Weske, Extracting Data Objects and their States from Process Models, in: Enterprise Distributed Object Computing (EDOC), IEEE, 2013, pp. 27–36. doi:10.1109/EDOC.2013.13.

[40] C. Combi, P. Sala, F. Zerbato, Driving time-dependent paths in clinical BPMN processes, in: Proceedings of the Symposium on Applied Computing, SAC '17, ACM, Marrakech, Morocco, 2017, pp. 743–750. doi:10.1145/3019612.3019620.

[41] H. Leopold, J. Mendling, A. Polyvyanyy, Generating natural language texts from business process models, in: International Conference on Advanced Information Systems Engineering, Springer, 2012, pp. 64–79. doi:10.1007/978-3-642-31095-9_5.

[42] W. M. van der Aalst, A. ter Hofstede, B. Kiepuszewski, A. Barros, Workflow patterns, Distributed and Parallel Databases 14 (1) (2003) 5–51. doi:10.1023/A:1022883727209.

[43] A. H. Morris, Developing and implementing computerized protocols for standardization of clinical decisions, Annals of internal medicine 132 (5) (2000) 373–383. doi:10.7326/0003-4819-132-5-200003070-00007.

[44] M. Dumas, M. Rosa, J. Mendling, H. Reijers, Fundamentals of business process management, Springer, 2013.

[45] F. Corradini, A. Ferrari, F. Fornari, S. Gnesi, A. Polini, B. Re, G. O. Spagnolo, A guidelines framework for understandable bpmn models, Data & Knowledge Engineering 113 (2018) 129 – 154. doi:10.1016/j.datak.2017.11.003.