Towards a Metamodel for Supporting Decisions in Knowledge-Intensive Processes

Sheila Katherine Venero
University of Campinas
Campinas, SP, Brazil
skvenero@ic.unicamp.br

Julio Cesar dos Reis
University of Campinas
Campinas, SP, Brazil
jreis@ic.unicamp.br

Leonardo Montecchi
University of Campinas
Campinas, SP, Brazil
leonardo@ic.unicamp.br

Cecília Mary Fischer Rubira
University of Campinas
Campinas, SP, Brazil
cmrubira@ic.unicamp.br

ABSTRACT

Knowledge-intensive processes (KiPs) cannot be fully specified at design time because not all information about the process is available prior to its execution. At runtime, new information emerges reflecting environment changes or unexpected outcomes. The structure of this kind of processes varies from case to case and it is defined step-by-step based on knowledge worker's decisions made after analyzing the current situation. These decisions rely on the knowledge worker's experience and available information. Current process management approaches still need to adequately address the complex characteristics of knowledge-intensive processes, such as their unpredictability, emergency, non-repeatability, and dynamism. This paper proposes a metamodel for representing KiPs aiming to help knowledge workers during the decision-making process. Domain and organizational knowledge are modeled by objectives and tactics. The metamodel supports the definition of objectives, metrics, tactics, goals and strategies at runtime according to a specific situation. Also, it includes concepts related to context and environment elements, business artifacts, roles and rules. The feasibility of our model was evaluated via a proof of concept in the medical domain.

CCS CONCEPTS

• **Applied computing → Business process modeling**; **Business process management systems**; • **Software and its engineering → Model-driven software engineering**;

KEYWORDS

knowledge-intensive process, business process modeling, case management, knowledge management, Process-Aware Information Systems, Business Process Management Systems

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297290

ACM Reference Format:
Sheila Katherine Venero, Julio Cesar dos Reis, Leonardo Montecchi, and Cecília Mary Fischer Rubira. 2019. Towards a Metamodel for Supporting Decisions in Knowledge-Intensive Processes. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3297280.3297290

1 INTRODUCTION

Knowledge-intensive processes (KiPs) are defined as business processes with critical decision-making tasks, and with a high dependence on knowledge worker's expertise and experience [2, 3]. The behavior of KiPs cannot be fully specified at design time because not all information about the process is available before its execution: new information emerges during their execution [17]. The planning for this kind of processes is done at runtime based on decisions. The course of action is determined step-by-step conforming to knowledge worker's decisions made after analyzing the contextual scenario and achieving intended goals [4, 17]. These decisions depend on the availability and content of data and knowledge elements, including rules and constraints[4]. Because of their complex structure, KiPs require substantial flexibility at design time and runtime [2, 4, 24].

Several data-driven solutions have recently appeared to support the complex characteristics of KiPs. These approaches provide a higher level of flexibility than imperative and declarative approaches because they are driven by data and not by an explicit control flow. Although these approaches are promising for managing KiPs, there is still a low maturity related to concrete tools and methods [4]. In order to support the decision-making process, some new approaches separate the decision logic from the process flow. This new modeling paradigm enhances the understandability and maintainability of models. However, this new paradigm brings other challenges regarding inconsistencies in later decision-process integration [9, 10, 13].

Di Ciccio, Marrella and Russo [4] identified the main KiPs' features and requirements to their management and execution. The authors defined an evaluation framework and analyzed the most representative and consolidated approaches in the Business process management (BPM) area that currently support KiPs. Their evaluation indicated that there is a lack of a holistic approach that satisfies all the identified requirements to manage KiPs. A tighter integration among data, processes, and users is required [4]. In addition, Marin *et al.* [16] analyzed the case management approaches using the evaluation framework proposed by [4] and concluded that the current case management approaches do not cover all KiPs requirements. This scenario demands extensions of models and methods to consistently integrate decisions, processes, cases, data, resources, and knowledge to fully satisfy KiPs features.

In this article, we propose a metamodel to capture KiPs' characteristics and to represent relevant elements for modeling and manage KiPs accurately. Our approach is focused on supporting knowledge worker's activities; more specifically, the decision-making process because decisions coordinate the structure of the KiPs. Hence, our metamodel allows modeling aspects such as resources, data, rules and constraints, goals, processes, collaboration, and the business environment into four major packages: data, knowledge, processes and decision in such a way that all those concepts are integrated. We believe that if we provide an explicit well-organized metamodel of KiPs concepts, the following aspects are addressed: i) the knowledge and data process are explicitly separated and externalized, allowing reasoning over them; ii) increase the transparency of the processes and data which might improve the data monitoring and process management; iii) explain the reasons (decision logic) for the taken decisions, a justification of choices; iv) increase the perception and understanding of knowledge workers so they can provide better-informed decisions and reuse former knowledge; v) gather tacit knowledge as making the decision logic explicit that could work for training inexperienced workers.

The created metamodel relies on a thorough analysis of the bibliography on KiPs. In order to structure the metamodel, we understood their significant characteristics and managed the key requirements; our metamodel was influenced by an analysis that indicated how the main existing approaches manage KiPs. The proposed metamodel is based on case management because, in our point of view, it is the paradigm that best attends the characteristics of the KiPs. We integrate the following perspectives: data, process, knowledge, and decision.

Data elements are organized into dynamic fine-grained artifacts and link them with atomic units of work (steps) to capture all interactions that involve data. So, the decisions are based on the data and also on the control flow, which make the process not only data-aware but also process-aware.Steps could also be linked to resources, so decisions can be also resource-aware. The decision logic is represented by a set of entities that decompose case situations and make further understandable the decision-making process. The collaboration during the decision making is represented through messages. To represent knowledge, our approach integrates features of strategic management such as goals, objectives, metrics, tactics, and strategies.

The feasibility of our metamodel was evaluated via a proof of concept in the medical domain. We analyzed the applicability of the metamodel for the diagnosis and treatment of patients, a real-life KiP. Medical processes are highly dynamic, complex, ad-hoc, and multi-disciplinary [22]; that's why they require flexibility and ad-hoc decision making.

The remainder of this paper is structured as follows. Section 2 presents fundamental concepts of KiPs and Section 3 shows the related work. Section 4 describes the proposed metamodel. Section 5 presents an application of our metamodel in the medical domain. Finally, Section 6 discusses the obtained findings, whereas Section 7 wraps up the concluding remarks.

2 BACKGROUND

Knowledge-Intensive processes (KiPs), or also called decision-intensive processes, are business processes whose behavior and execution are heavily dependent on users performing collaborative knowledge-intensive decision-making tasks. KiPs are inherently people-centric because their activities are mainly performed by knowledge workers who make autonomous decisions using their experience and expertise. These processes are knowledge-, information- and data-centric processes. Indeed, KiPs are highly dynamic and knowledge workers' decisions contribute to the definition of the best course of action. Knowledge workers can instantiate and concretize specific procedures, or contextually select and compose an appropriate plan during process execution. Recent literature indicates the need of more freedom to knowledge workers manipulate process activities, and looseness of process execution [4, 23, 24].

Di Ciccio, Marrella and Russo [4] defined eight key representative characteristics of KiPs: Knowledge-driven, Collaboration-oriented, Goal-oriented, Constraint- and rule-driven, Event-driven, Unpredictable, Emergent, and Non-repeatable.

KiPs are Knowledge-driven because the content and availability of the data and knowledge (explicit and tacit) drive the human decisions which directly routes the flow of the process. Usually, KiPs are performed in a collaborative multi-role-user environment, where users create, share, and transfer information to innovate and manage processes. KiPs are goal-oriented because during its execution goals and milestones need to be achieved. Such goals may be set at the beginning of the process, but they can change according to the situation. As in any process, the rules and constraints have to be fulfilled. Events regarding activity completion, data, knowledge, context and environment may change human decisions and consequently change the process evolution. In addition, Kips are unpredictable and emergent because some new information can appear at runtime changing the curse of actions. So, they are defined, step-by-step, as soon as the information is available. KiPs depend on the current situation and context-specific elements, so it cannot be entirely modeled at design time. As the conditions vary in every process instance, they can hardly be repeatable.

To support those complex characteristics, Di Ciccio, Marrella and Russo [4] defined 25 requirements divided into seven categories: Data, Knowledge Actions, Rules and Constraints, Goals, Processes, Knowledge Workers and Environment. In this sense, process management systems should provide support for the definition, evolution, monitoring, and analysis for each category.

3 RELATED WORK

Several data-driven approaches have appeared in the last years to support KiPs. Data-centric approaches use complex data structures or information models to identify activities and domain-relevant object types, relationships and states. The course of action is guided by values on data objects and not for activity completion. Case management approaches represent activities through forms and focus on individual cases driven by availability, changes, and evolution of values for data objects and their dependencies. Artifact-centric approaches are driven by a case file which represents all data and context related to a case at runtime.

The *PHILharmonicFlows Framework* [15] uses an object-aware paradigm which provides a form-based data and process-oriented views. It uses a relational data model for the definition of object types and their attributes, users, roles, activities (worklist-based); and their relations and interdependencies. The framework does not explicitly allow goal modeling, whereas declarative and procedural modeling can be combined. Changes in attributes of objects guide the process execution, but late modeling is not explicitly supported. PHILharmonicFlows framework partially supports the decision-making process, and it does not support collaboration among users.

SmartPM [18] allows the representation of data and knowledge elements associated with a process schema (process tasks, rules, constrain, goals). In their work, data is represented through data objects; data types are related to domain objects and users. Roles and capabilities are also supported, but the data access is not clearly explained. It allows late modeling of process activities. SmartPM automatically adapts processes at run-time using Artificial Intelligence techniques. It helps users' decisions through the late selection at runtime. In their work, collaboration among users is not provided.

Artifact-GSM [12] uses a artifact-centric paradigm and is centered on a information model. Data and status attributes are used to verify the progress of an artifact instance. It allows the definition of tasks, goals, rules, internal and environment events through stages, guards, milestones, and sentries. Roles definition is based on an authorization model. This proposal does not support late modeling. User's decisions support process coordination and data evolution through declarative and rule-driven model. Collaboration is not explicitly treated.

The most representative Adaptive Case Management (ACM) approach is the Case Management Model and Notation (CMMN)[7] standard released by the Object Management Group (OMG) [21]. CMMN defines an information model with unstructured and structured data. Changes in the information model generate events that can enable tasks, stages, or milestones. It allows late modeling for data and actions which gives more flexibility during execution. It partially supports modeling goals, external events, rules, and constraints. This proposal allows the definition of resources, roles, and capabilities. As it is a modeling notation, collaboration is implemented in the running environment [4, 16].

Along with CMMN, the Decision Model and Notation (DMN) [8] standard, a declarative decision language, separates decision from the process model. DMN divides decisions into two levels: the decision requirement level and the decision logic level. The model represents the decision logic in decision tables. DMN aims to be used along with the BPMN (Business Process Modeling Notation) and CMMN. However, there are issues regarding the integration of decisions into the entire process as required in KiPs and not only in local decision points.

KIPO ontology [5], a domain-independent ontology, addresses concepts related to knowledge, decision-making, and collaboration. It provides means for representing activities, goals, rules, roles, and decisions. The KIPO ontology does not explicitly show the data object integration with the activities. It is unclear whether late modeling is allowed.

Recently, Mertens *et al.* [19] proposed a metamodel for modeling KiPs, called *DeciDeclare*. This employs a mixed-perspective process language that integrates the functional, control-flow, data and resource perspectives into a single metamodel based on DECLARE [25]. Although this approach allows data-awareness and resource-awareness, it does not explicitly support neither goal nor decision modeling. Their metamodel does treat the collaboration aspect.

These approaches support KiPs in a better manner than traditional activity-centric approaches because they are data-driven. Each approach supports in a better way some requirements and partially supports others. However, none of them cover all defined requirements for KiPs [4]. Some inherent characteristic of KiPs, such as its emergency and unpredictability, are neglected since none of the approaches satisfactorily allows late modeling. Collaboration among users is also another caracteristic left out. Modeling decision is not explicitly addressed or is partially supported by most of the approaches.

Our proposed metamodel mapped concepts from strategic management and explicitly integrates decision with tasks, resources, data, and Knowledge elements. Domain and organizational knowledge elements are organized following the strategic management concepts. Objectives, rules, and metrics (key performance indicators) are first formulated for the organization to pursue. Former knowledge about policies and plans are modeled using the organization resources. Then, dynamic business artifacts can be modeled and updated at runtime. The metamodel allows late modeling of processes and goals via strategic decision modeling, which stands for the key originality in this investigation. Our primary objective is to support Knowledge workers during the decision-making process. We believe that our metamodel might help knowledge workers to have better critical thinking and to formulate, organize, and implement goals taking on consideration the environment in which the organization operates. Since the course of KiPs depends on knowledge workers' decisions, an adequate representation of how decisions are made is crucial for understanding the reasoning of knowledge workers to learn from their experience.

4 METAMODEL FOR KIPS

Our metamodel defines entities and attributes for representing a knowledge-intensive process independently from the domain. We reused and integrated concepts existing in the literature regarding process management. The model was mostly inspired by BPMN[6], CMMN[7], DMN[8] and KIPO Ontology [5].

Giving the high degree of flexibility required for handling KiPs, we explore a declarative modeling paradigm because it allows us to postpone decisions about the process control flow until its deployment or even its execution. Deferring decisions during execution increases the flexibility of processes execution and deals with the uncertainty regarding the exact specification of the process [23]. At runtime, tactic templates and activities are dynamically selected according to a given situation. Our metamodel supports late selection and late modeling and composition. The metamodel is organized into four main packages:

- **Case Package**: It defines the base structure of the metamodel, a Case. A case definition represents a structure based on artifacts, that is, documents used by the organization.

The internal structure of a case has a context, a status, and environment. A case instance presents a behavior defined at runtime.

- **Control-flow Package**: The behavior of a case is composed of a set of activities to handle a case. Activity definitions are made in a declarative way. Our metamodel refines the granularity of the work unit; an activity is composed by a set of tasks; a task comprises a set of steps. Step definitions are associated with one attribute of an artifact, one resource and one role type at most.
- **Knowledge Package**: It stands for a knowledge base representing domain and organizational rules, norms, guidelines, best practices and standards. This knowledge is encoded in ECA rules (Event-Condition-Action rules), tactic templates, objectives and metrics.
- **Decision Package**: It represents the structure of a collaborative decision-making process performed by knowledge workers using the knowledge base. First, to align the process with the changing needs, we decompose a case instance's situation to have a big picture of the current context and environment. Then, we define long-term goals that guide plans with short-term objectives. This approach is based on the strategic decision-making.

In the following, we detail the elements of each package and describe its key features.

4.1 Case Package

Figure 1 presents the structure of the case package. A *case* is defined by its *context*, *status*, *environment*, which in turn are described by the organization *artifacts*.

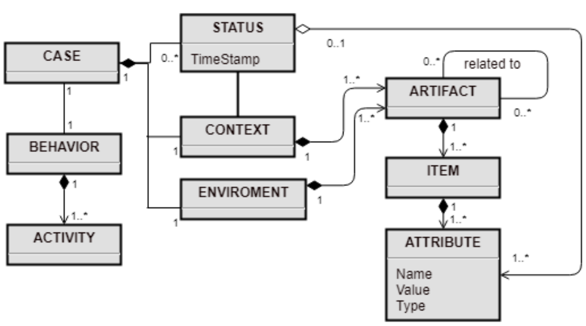


Figure 1: Case Package

The **case** relies on the definition of *artifacts*. We consider an **artifact** as a physical document used by workers in the organization for performing their daily activities. For instance, medical records are daily used by nurses and physicians. An *artifact* is a data object which stores relevant information of a *case*. An *artifact* is composed by a set of *items*, which represents a section in a document (*cf.* Figure 2). Each *item* has atomic and structured *attributes*, which stand for data values to be recorded in a document. At runtime, data values are stored in each *attribute*. Data stored in an *artifact* may serve as input for some *activities*, becoming a precondition for those *activities*. The logical relationships among *artifacts* represent the structure of the business domain.

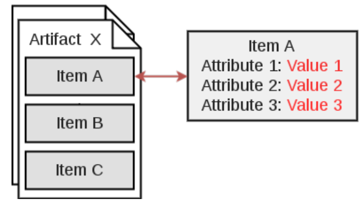


Figure 2: Structure of an *artifact*

The **context** represents relationships among all the *artifacts* considered relevant to the case. The **status** of a *case* can be seen as a set of critical values related to *attributes* of the *artifacts* inside the *context* model. These *attributes* are defined prior the *case* instantiation.

The **environment** of a *case* is characterized by a set of relevant variables about physical, social, and others conditions in which the *case* occurs. It is represented by a set of relevant *artifacts*, in which *attributes* represent the environment variables.

The **behavior** of a *case* is the execution trace of the *activities* performed to handle the *case*, which is defined at runtime.

Artifacts can be evaluated at any time according to the values of their attributes. Some other variables can appear at runtime, so new artifacts or new attributes should be added into the definitions of context and environment models. The definition of basic *artifacts* must be declared before the execution of case instances. However, at runtime new artifacts can be modeled.

4.2 Control-Flow Package

As KiPs are a type of a process, we defined basic elements for this purpose such as **activities**, **roles**, **actors**, and **resources**. Figure 3 presents the elements of this package.

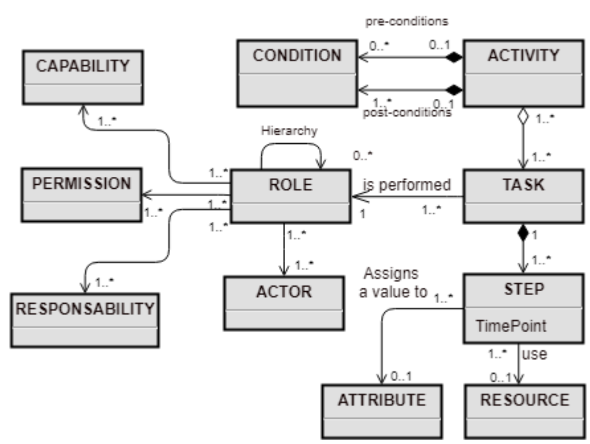


Figure 3: Control-Flow Package

An **activity** is composed of one or more *tasks*. A **task** comprises a set of *steps* that are performed by one actor of a specific *role*. **Steps** are logical partitions of a *task* and may depend on each other representing a pattern. A step represents an atomic unit of work performed at a single time point. Some *steps* may require data values as an input and may produce a data value as output. *Steps* may be associated at most with a single *resource* and attribute of an *artifact* in the *context* of a *case*. *Activities* and *tasks* may require the definition of pre- and post- conditions and other scheduling constraints. For instance, for defining sequential *tasks* or *steps*. If a condition is not satisfied, the unit of work cannot be activated. We increase the granularity of *activities* because it makes easier the execution, improves the visibility of data values and allows collaborative activity execution.

Roles are functions performed by actors, and they are defined through capabilities, responsibilities, and permissions. *Role* definitions may have some hierarchy among them depending on the domain. An actor may be associated with many *roles*. An **actor** represents professionals involved in the execution of *tasks*. They can be specialized in human and non-humans actors. A **resource** is an item used to support the execution of *steps*.

During the process execution, *activities*, *tasks*, and *steps* change states according to basic transition schemas. Figure 4 presents the basic states for an *activity* and a *task* by involving: initiated (concrete *activity*/*task* has been created), running (the execution has started, inner *tasks* or *steps* may be started), active (one or more *task*/step has been started), suspended (no *task* or *step* is started until the root *activity*/*task* has returned to running state), completed (the *activity*/ *task* has fulfilled the conditions for completion, post-conditions reached), and terminated (the execution has stopped because of an abnormal cause).

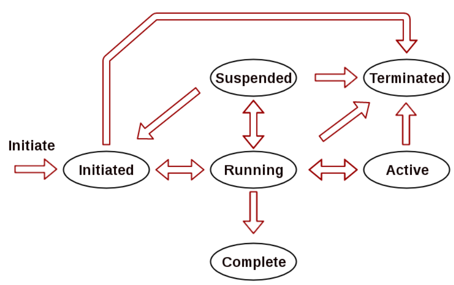


Figure 4: State diagram of activities and tasks

As *steps* are atomic work units, they have to be entirely executed or "rolled back" to a previous step. Figure 5 presents the state transitions for a step which includes: inactive (a concrete step has been created but is not activated because not all preconditions have been met), active (processing), suspended, completed successfully, and failed. The current state of a step is strongly connected to the state of *task* parent. Similarly, the state of a *task* is related to the state of *activity* parent. The states of *steps*, *tasks* and *activities* are synchronized with the states of subordinated *activities*, *tasks* and *steps*. The state transition schemas were adapted based on the state transitions of the workflow enactment service [11].

During the implementations of the *tasks*, concrete *actors* perform *steps* and may produce some data values; steps would be only completed when required data values are provided by an explicit commitment. *Steps* should be connected to interactive forms representing *artifacts*. One form is associated with many *steps*. The produced data value on a step acts as the driver for the control flow; it is considered as a transition for connecting *tasks*. After the

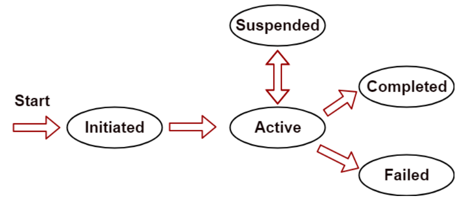


Figure 5: State diagram of steps

completion of a step, the produced data value is analyzed to verify correctness to the defined *business rules*, *objective* and *metrics*.

4.3 Knowledge Package

In order to create possibilities of supporting runtime coordination of *activities*. We modeled essential elements for representing explicit knowledge in the following entities: **objectives**, **metrics**, **tactics**, **business rules**. Figure 6 presents the elements involved in the knowledge package.

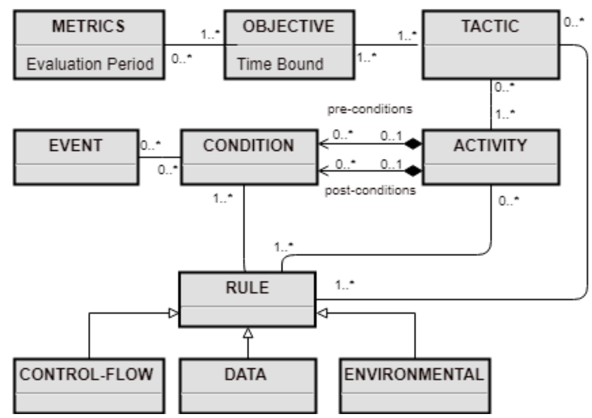


Figure 6: Knowledge Package

An **objective** is an operative goal, specific, measurable, and quantifiable. It has a reference time bound for being reached and may have some domain *metrics* associated to it. A **metric** is a quantifiable measure used to track and assess the status of an *objective*, such as key performance indicators (KPI) or key risk indicators (KRI). The *metrics* help the detection of deviations and unexpected process evolution, and give a view of the achieved progress in reaching an *objective*. *Metrics* are measured during or after execution, an evaluation period has to be defined.

In this proposal, an *objective* is achieved through different *tactics* (*cf.* Figure 7). A **tactic** is a detailed to-do activity list, a sequence pattern representing best practices and guidelines, for pursuing an *objective*. A *tactic* is defined as a rule-based combination of *activities*. *Tactics* may evolve and be created at runtime, *e.g.*, removing, adding, updating *activities*. In our model, they serve as *tactic* templates to be instantiated by knowledge workers to deal with a situation in a *case*.

Rules are modeled as ECA (Event-Condition-Action) rules. When an event is triggered, a condition is evaluated, and an effect may occur. An **event** is triggered by the state transition of *activities*,

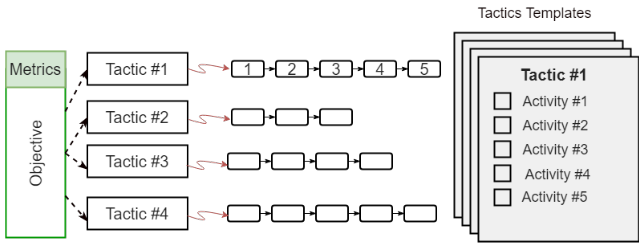


Figure 7: Relations between objective and tactics

tasks, *steps* or by a data value update. *Rules* can be specialized according to some aspects: control-flow, data, and environmental. The *control-flow rules* describe constraints, permissible way of connecting *activities*. *Data rules* describe reference values for business and process data. *Environmental rules* describe external regulations, norm and policies that may affect the execution of *activities*.

4.4 Decision Package

The structure of KiPs is driven by decisions made at runtime. This part of the metamodel defines an explicit logic for making a decision based on strategic management. Figure 8 presents the entities involved in the modeling of a strategic decision.

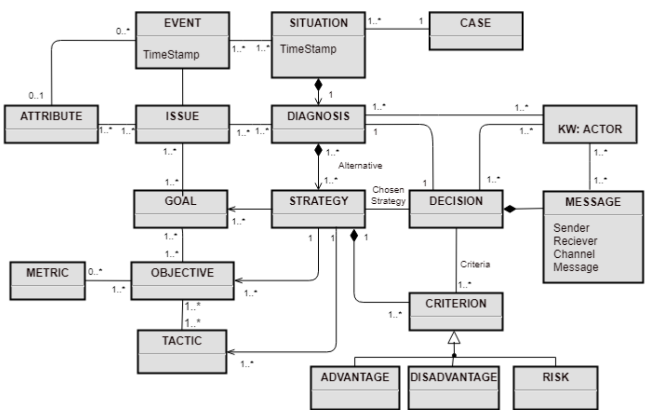


Figure 8: Decision Package

A **situation** is a tuple (E, C, St): where E is the set of events that triggered the situation; C is the associated *case* instance, which is the set of attribute values contained in artifacts of the *context* and the *environment* at a given time; and St is the *status* of C at a given time. The status St may be either the initial assigned values, or the values resulting from other decisions. Each event of E may generate zero or more issues.

An **issue** is an *attribute* value which is considered critical or unsatisfactory for the case instance. A **diagnosis** D is a set of *issues* $Is_1, Is_2, \ldots, Is_k$. The *diagnosis* defines the nature of the challenge, which will lead to a ad-hoc decision. Knowledge workers must provide a diagnosis.

Given a diagnosis, there can be several alternative *strategies* to solve the *issues*. A **strategy** is defined as a high-level plan with the aim of solving the identified *issues*. Each *strategy* may have associated to one or more *criteria*, each of which represents an *advantage*, a *disadvantage*, or a *risk*. Knowledge workers must choose the adequate strategy. They come to a conclusion about the best way to deal with identified *issues* based on both the *strategy's criteria* and other ad-hoc *criteria*. Once the "best strategy" is identified, a decision is made. A **decision** for a given diagnosis is a pair (S, Cr), where S is the chosen strategy, and Cr is the set of criteria on which the decision was based.

The approach of an strategy is based on achievement of *goals* through *objectives* and *tactics*. *Goals* are oriented to solve one or more issues identified in the *situation*, and can be reached through the achievement of *objectives*. A **goal** is defined as a pair $goal = (Is(g), Ob(g))$, where $Is(g)$ is a subset of issues, and $Ob(g)$ is a set of objectives. In turn, each objective is achieved by an associated *tactic*. More formally, a **strategy** for a diagnosis D is a set of goals $g_1, g_2, \ldots, g_k$ such that the union $Is(g_1) \cup Is(g_2) \cup \ldots Is(g_k)$ is a superset of D.

Figure 9 presents an example of one *strategy* to solve four *issues*. The first *goal* aims to solve *issue* 1 and it is attached to three *objectives* organized according to priority. The second *goal* aims to solve *issues* 2 and 3 and it is attached to one *objective*. The third *goal* aims to solve *issue* 4 and it is attached to two *objectives*. Each *objective* becomes a step toward reaching a *goal*. Each *objective* is reached through one *tactic*.

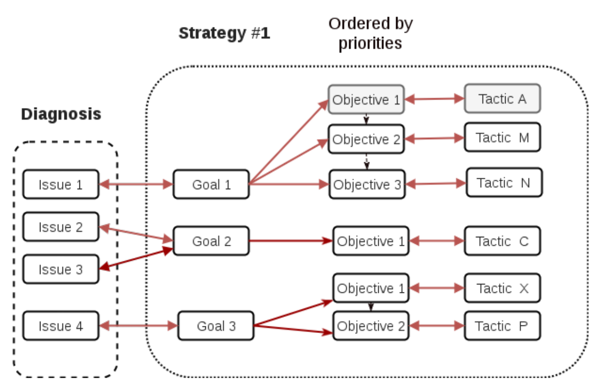


Figure 9: Diagnosis with a suggested strategy

The achievement of an *objective* is a necessary milestone to make progress toward its respective *goal*. An *objective* may depend on another *objective*, and thus they must be ordered by priorities. Similarly, the achievement of a *goal* is a milestone for the success of a *strategy*, and must also be ordered by priorities. Each *objective* should be related to one recommended *tactic* from the knowledge base or with some new *tactic* composed at runtime. Knowledge workers can choose a *strategy template* used in a past case instance that solve a similar *situation* or create a new one based on *tactic templates*.

Collaboration among knowledge workers is usually required due to the difficulties involved in the decision-making process. Messages are exchanged among decision makers for creating a new strategy or for reaching a consensus about the best solution that produces the desired result given a specific *situation*. A **message** is an exchange of information transmitted by a channel between knowledge workers playing roles of senders and receiver. Through *messages* tacit and explicit knowledge of the knowledge workers is exchanged. In this perspective, it is very important to store that information to learn from Knowledge workers experience and knowledge.

5 APPLICATION

We present an evaluation of the proposed metamodel as a Proof of Concept (PoC) in the medical domain. More specifically, we address the case of diagnosis and treatment of patients. The main goal of this evaluation is to validate whether the proposed metamodel is able to represent a real-life KiP and to provide appropriate information to knowledge workers for taking right decisions at runtime.

The diagnosis and treatment of patients is a typical Knowledge-intensive process; in fact, it requires lots of knowledge and reasoning. Knowledge workers have to diagnose the health problems and give treatment, so this process strongly depends on their experience and expertise. The scenario of the treatment of the patient involves deciding which procedures and examinations are necessary in a patient-specific case, depending on the current state of health of the patient, contraindications, possible side-effects or risks from their decisions, cost and time of the procedures, resource availability, and so forth.

In addition, not all patients with the same disease receive the same treatment. Each patient has a unique health state and responds differently to a particular treatment. The therapeutic procedures must be scheduled in a coherent order, taking into account pre-conditions and post-conditions, dependencies, and the distance in time between tasks. The decisions about next steps require dynamically scheduling during the treatment. Finally, the diagnosis and the treatment of a patient usually involves several units in a hospital, so it has a high degree of collaboration among participants.

In order to evaluate the metamodel, we created a case model and then executed a case instance. Before the case instantiation, it was necessary to make definitions for the medical domain, more specifically, for a hospital environment. Therefore, we identified relevant information for a hospital and then mapped them as entities of the metamodel, as follows.

- Case Definition → Treatment of a particular patient
  - Case Context → Electronic Health Record (EHR) (set of different kinds of artifact), Physical Examination, Evolution of the Treatment
  - Case Status → Vital Signs and Main Complaints
  - Case Environment → Information regarding a hospital's environment: *e.g.*, Medicine Inventory, Equipment, Stock of Resources, Medical Staff, Federal Regulations.
- Behavior → medical procedures performed on a patient (Ad-hoc treatment, defined at runtime)
- Tactics, objectives and metrics → Clinical guidelines, international standard procedures for diseases, established organizational protocols of the specific organization.
- Activities, Task, and Steps → Therapeutic activities.
- Roles → Information regarding a hospital's organization: Doctors, Nurses, Allied health professionals (dietitians, occupational therapists, pharmacists, physiotherapists, podiatrists, speech pathologists, and so forth), Support staff (clinical assistants, patient services assistants, porters, volunteers, ward clerks)
- Rules → Normal health reference ranges. The normal ranges for a person's vital signs vary with age, weight, gender, and overall health.

The next step was to populate the knowledge base of our model. To instantiate a medical case, we assumed that the medical information was already stored in the knowledge base regarding rules, tactics, objectives, metrics, activities, task, steps, and a hospital environment. We also assumed that an inference engine was available to make suggestions for knowledge workers regarding critical issues in a situation and potential tactic templates solutions.

Therefore, we simulate a case instance, a basic diagnosis and treatment example. To this end, consider the following medical scenario in the emergency room. The example was created based on nursing guidelines such as [1, 20].

Joseph, Man, 49 years old, married, two children, admitted on 20/06/2018 at 2h30. On admission, he was febrile at 38° C. He reported that the temperature was increasing during the day.

Figure 10 presents the instantiation of a *situation*, labeled as S001. This situation was triggered after attributes of the artifacts in the context file on the case instance, labeled Case001. They were filled out generating a set of events. For example, it was inserted the *attributes* "Problem" and "time course" in *item* "Chief Complain" that belongs to the "Present Illness History" *Artifact*. Figure 10 highlights attributes that were filled conforming to the provided information.

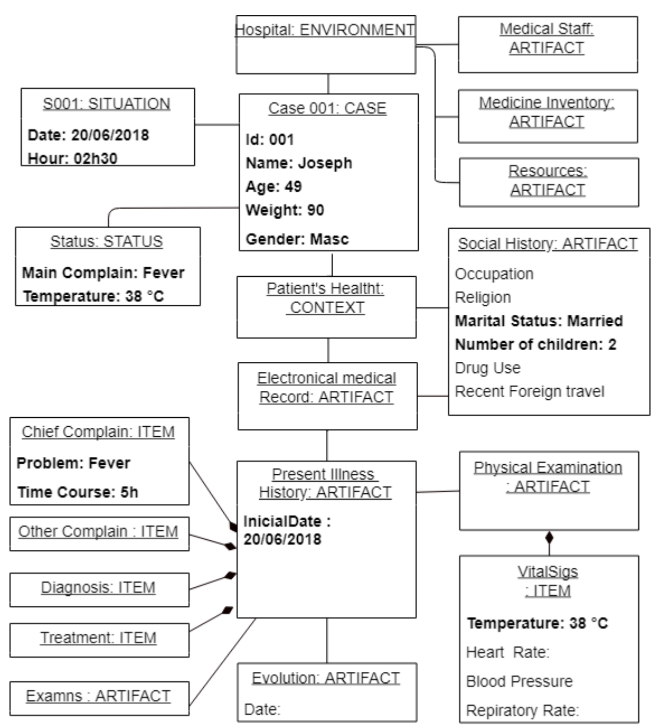


Figure 10: Case Instance: Status, Context and environment artifacts

This case instance was assigned to doctor Hans, which is available according to the information on the medical staff artifact. Analyzing the current data, the system signalizes that the data value for the attribute "temperature" inside the item "Vitals Signs" in the Artifact Physical Examination, has a violation of the *Data Rule: Normal body temperature rate → 36.5 °C – 37.5 °C*. Immediately, an instance of diagnosis, labeled D001, is created based on the knowledge base with the issue IS001: High value for body temperature. Then, Doctor Hans accepts the diagnosis and set a goal S001G1 to solve the issue IS001. With this commit, the system performs a query for operational objectives to reach the goal. After making inferences, the system creates a suggestion of a strategy ST001, shown in Figure 11. The goal S001G1 is realizable through the objective OBJ150 and tactic TAC150 which has seven activities. Activities are oriented to the achievement OBJ150 objective. The fulfillment of objective OBJ150 is evaluated through the metric ME001.

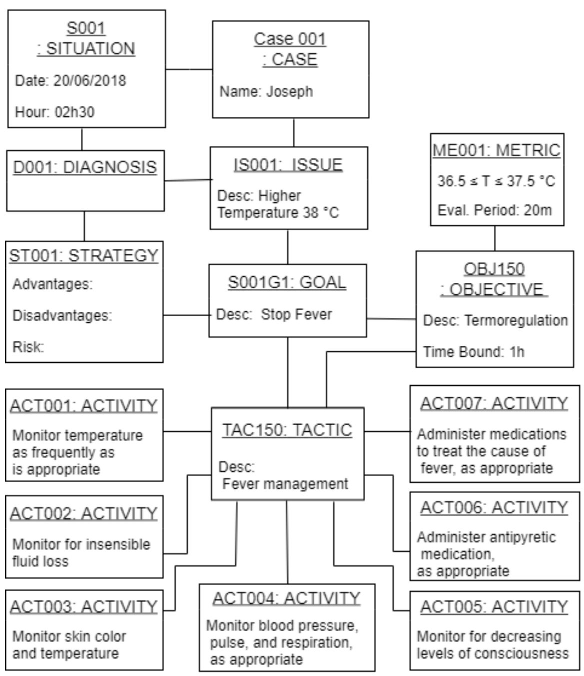


Figure 11: First suggested strategy according to symptoms and signs

With this suggestion, Dr. Hans selects the activities that she believes to correspond to that case. The system shows a list of antipyretic medications, and she recommends to give the patient a dosage of Paracetamol 500g and monitor patient's temperature. Figure 12(a) shows the taken decision, the chosen strategy ST001 and criteria CR001. Activities ACT001 and ACT006 were chosen modifying the original tactic. The ACT001 has one task, performed by a nurse, which has three steps, Step 3 is related to the attribute temperature of the item vital signs at Evolution artifact. The activity ACT006 has two tasks, the task01 is performed by a doctor, and the task02 is performed by a nurse. Steps are attached attributes of artifacts on the case. Having taken the decision, the implementation of the strategy starts. All linked *activities*, *tasks* and *steps* are automatically initiated and ordered according to the defined priorities and dependencies. Concrete *actors* and *resources* are automatically assigned to *tasks* according the hospital environment variables.

During the execution of the activities, new events occur. In this example, new symptoms are affecting the patient, updating data values of artifacts.

At 3h00 hours, Joseph reports new symptoms abdominal pain and mild diarrhea.

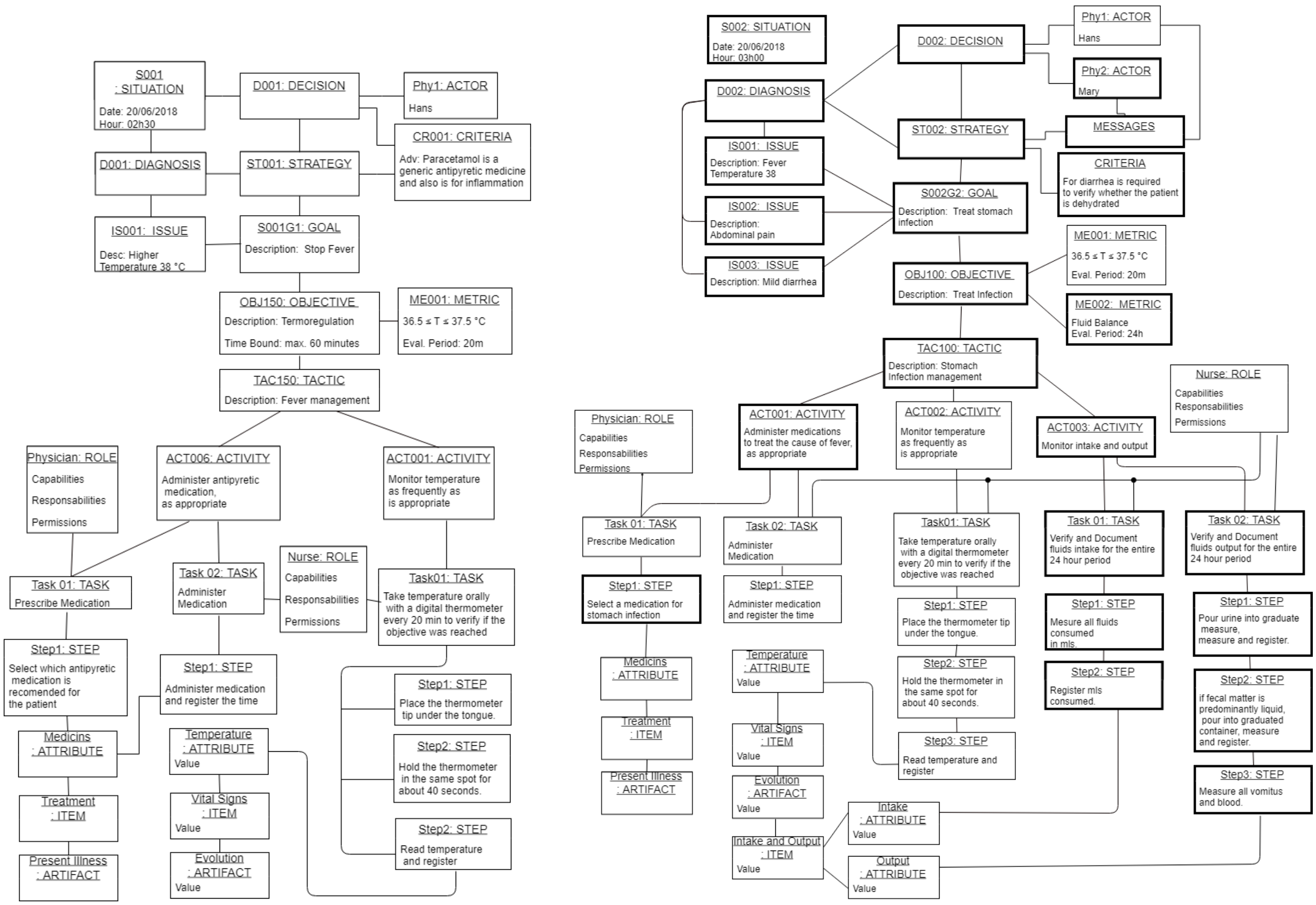
Thus, the addition of new complaints in the item "other complains" of the artifact "Present Illness History" generates a new event. Such new event triggers a new situation S002 (see Figure 12(b)), and two more issues are added to the diagnosis. With this new information, Dr. Hans explains the situation to Dr. Mary, and after some exchange of messages, they concluded that those symptoms show a stomach infection. The system shows tactics for a stomach infection, Dr. Hans selects appropriate activities for stomach infection and prescribe Ciprofloxacin 500mg every 12 hours for five days. This strategy is shown to Dr. Mary that now is collaborating for the creation of the new strategy. Moreover that treatment, she wants to monitor fluid intake and output during the next 24 hours. Therefore, the decision of the treatment is based on criteria CR002, and it was taken by both doctors, using the new formulated strategy ST002. Figure 12(b) shows the new strategy ST002. There are three issues in diagnosis D002 and one goal S002G2 to achieve. The goal is realizable by objective OBJ100 and tactic TAC100. Tactic TAC100 has 3 activities.

After the strategy to treat the patient has been confirmed, the structure of the process is modified according to the activities in tactics, ready to be implemented. Continuously, the process execution has to be monitored to verify the existence of new, unexpected situations and also the fulfillment of objectives, goals and finally of the strategy as a whole. The adaptation of strategies, tactics, and goals can be repeated several times throughout case lifetime. If the goals of the strategy are achieved, the template of the strategy is stored in the knowledge base as a successful strategy. Similarly, templates of new *tactics* are stored.

6 DISCUSSION

The support for modeling and execution of KiPs remains an open research challenge. In this paper, the proposed a metamodel focused on the representation of KiPs aiming to support knowledge workers during the decision-making process. The proposed metamodel covers all relevant KiPs characteristic and requirements. Our metamodel shows an explicit integration of the data, domain and organizational knowledge, rules, goals, and activities.

The Case Package allows the representation of data, the process data and environmental data. This data is modeled by business artifacts, following the artifact-centric paradigm. Artifacts represent real-life documents, so they are presented to users as interactive forms in a similar way as real documents, which facilitates the understanding. Besides, the fact of data being modeled by business artifacts facilitates the insertion of new emergent information, which enables adding new artifacts to the context file of the case. The metamodel allows the modeling of the organizational environment where the case is executed. As it is also modeled through artifacts,



(a) 1st Decision, refinement of tactic and strategy

(b) 2nd Decision, new strategy conforming to new events

Figure 12: Chosen Strategies

it enables to add information at runtime. So, our metamodel allows late data modeling.

An essential part of the metamodel is the representation of knowledge. Our model captures both explicit knowledge and tacit knowledge. Explicit knowledge such as domain and organizational guidelines are captured by rules, objectives, tactics, and metrics. Objectives are formulated through metrics defining target values and evaluation periods which permits the continuous process progress evaluation. Tactics encode domain and organizational rules and practices. They serve as process templates to be used to achieve an objective. Objectives are realizable by tactics, and tactics are composed of operative activities.

In addition, the metamodel allows collaboration during the decision-making process through message exchanges, a communicative interaction. Our proposal provided means for storing each step during the decision-making process. During the creation of strategies and tactics, the workers exchange messages discussing and explaining in one way or another their reasoning. These interactions are tracked by messages as elements of the metamodel. Thus, the information system can capture the knowledge workers' tacit knowledge encoded in messages. The management and treatment of that information enables learning from knowledge worker experience and tacit knowledge.

Considering the process perspective, the metamodel allowed modeling resources, roles, actors, activities, tasks, steps. The metamodel enabled the representation of roles based on capabilities, responsibilities, and permissions. These definitions are relevant because they guide the allocation of roles to tasks and they restrain the data access during or after execution. Activities can be modeled in a declarative way based on rules and constraints giving more flexibility to the metamodel. Activities may comprise different tasks, and each task is assigned to one role. Thus, activities can be performed collaboratively by one or more actors. A task is logically divided into steps, which allows the better management of data entry on the artifacts. The state transitions for activities, tasks, steps make possible the monitoring of the course of action and data value updates inside the context and environment artifacts. That enables runtime activity coordination and process analysis. Every step, task, and activity performed contributes to the progress of the process and the success of a strategy. Metrics were defined to asses the achievement of objectives and goals. Metrics are measured at runtime according to their respective evaluation period which also allows controlling KiPs execution. In this sense, our metamodel is not only data-aware but process-aware.

The metamodel supports the decision-making process by the creation of strategies according to new circumstances and emergent events. It helps knowledge workers in the decision-making process, by providing a formalized representation of explicit knowledge regarding past cases, strategies, objectives, and tactics. We proposed a formal representation of how decisions can be made by using principles of strategic management. The strategy structure is modeled at runtime by goals, objectives, metrics and tactics templates. The strategy defines the coordination of the process. The metamodel provides a good organization for retrieving tactics that match similar objectives to goals which facilitate the reuse of knowledge.

We showed the usage of the proposed metamodel in a medical scenario indicating that the metamodel allows the representation of a real-world KiP, such as diagnosis and treatment of patients. Withal in the developed proof of concept, we used tactics and activities based on nursing guidelines [14, 20]. In this sense, it was not possible to confirm the precise adequateness of the information about the strategies, objectives, tactics, and activities. The situation modeled must be further assessed in a thorough hospital situation. In addition, we need to explore further real-life application scenarios to evaluate additional aspects of the metamodel.

7 CONCLUSION

This paper proposed a metamodel for representing knowledge-intensive processes (KiPs). The metamodel has been defined by analyzing the concepts related to business process management presented in literature, including case management, knowledge management, and strategic management. After presenting our metamodel, we demonstrated its application to a realistic knowledge-intensive process in the medical domain.

The metamodel aimed at supporting the decision-making process providing a logical structure for organizing and maintaining the process and knowledge data. Hence, besides process modeling, semi-automatic techniques can be used for process mining, simulation and conformance checking of past and present process executions.

Further directions for this work consist in: i) the development of a concrete syntax for the modeling language; ii) a comprehensive evaluation of the approach with data from real medical processes as well as with experts (process owners and knowledge workers); and iii) a full implementation of an infrastructure to manage KiPs based on our metamodel to provide support for the definition, monitoring, execution and analysis of the processes. To this end, we will base our work on ontologies and semantic reasoning. Finally, we plan to understand the applicability of the proposal to different domains.

ACKNOWLEDGMENTS

This work is partially supported by CAPES and CNPq scholarships, by CNPq PQ Grant #306089/2017-3, and by the São Paulo Research Foundation (FAPESP) (Grants #2017/02325-5 and #2017/21773-9)[1].

REFERENCES

[1] Howard K Butcher, Gloria M Bulechek, Joanne M McCloskey Dochterman, and Cheryl Wagner. 2018. *Nursing Interventions classification (NIC)-E-Book.* Elsevier Health Sciences.
[2] Thomas Davenport. 2005. *Thinking for a Living. How to get better performance and results from knowledge workers.* Boston: Harvard Business School Press.
[3] Thomas Davenport and Laurence Prusak. 1998. *Working Knowledge: How Organizations Manage What They Know.* Harvard Business Press.
[4] Claudio Di Ciccio, Andrea Marrella, and Alessandro Russo. 2015. Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches. *Journal on Data Semantics* 4, 1 (2015), 29–57.
[5] Juliana Baptista dos Santos França et al. 2015. KIPO: the knowledge-intensive process ontology. *Software & Systems Modeling* 14, 3 (2015), 1127–1157.
[6] Object Management Group. 2011. Business Process Model and Notation (BPMN). formal/2011-01-03. Version 2.0.
[7] Object Management Group. 2016. Case Management Model and Notation (CMMN). formal/16-12-01. Version 1.1.
[8] Object Management Group. 2018. Decision Model and Notation (DMN). dtc/2018-06-01. Version 1.2.
[9] Faruk Hasić et al. 2017. Challenges in refactoring processes to include decision modelling. In *International Conference on Business Process Management.* Springer, 529–541.
[10] Faruk Hasic, Linus Vanwijck, and Jan Vanthienen. 2017. Integrating processes, cases, and decisions for knowledge-intensive process modelling. In *International Workshop on Practicing Open Enterprise Modeling, CEUR.*
[11] D. Hollingsworth. 1995. *Workflow Management Coalition - The Workflow Reference Model.* Technical Report. Workflow Management Coalition.
[12] Richard Hull et al. 2011. Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events. In *Proceedings of the 5th ACM international conference on Distributed event-based system.* ACM, 51–62.
[13] Laurent Janssens, Ekaterina Bazhenova, Johannes De Smedt, Jan Vanthienen, and Marc Denecker. 2016. Consistent integration of decision (DMN) and process (BPMN) models. In *Proc. of the CAiSE'16 Forum, at the 28th International Conference on Advanced Information Systems Engineering (CAiSE 2016)*, Vol. 1612. 121–128.
[14] Marion Johnson et al. 2012. *NOC and NIC Linkages to NANDA-I and Clinical Conditions: Supporting Critical Thinking and Quality Care.* Elsevier Mosby.
[15] Vera Künzle and Manfred Reichert. 2011. PHILharmonicFlows: towards a framework for object-aware process management. *Journal of Software: Evolution and Process* 23, 4 (2011), 205–244.
[16] Mike A Marin, Matheus Hauder, and Florian Matthes. 2015. Case management: an evaluation of existing approaches for knowledge-intensive processes. In *International Conference on Business Process Management.* Springer, 5–16.
[17] Olivera Marjanovic and Ronald Freeze. 2011. Knowledge intensive business processes: theoretical foundations and research challenges. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on.* IEEE, 1–10.
[18] Andrea Marrella, Massimo Mecella, and Sebastian Sardina. 2014. SmartPM: An Adaptive Process Management System through Situation Calculus, IndiGolog, and Classical Planning.. In *Poce. of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR'14).* 518–527.
[19] Steven Mertens, Frederik Gailly, and Geert Poels. 2017. Towards a decision-aware declarative process modeling language for knowledge-intensive processes. *Expert Systems with Applications* 87 (2017), 316–334.
[20] NANDA International. 2014. *Nursing diagnoses: definitions and classifications.* John Wiley & Sons.
[21] Object Management Group. [n. d.]. https://www.omg.org/. Accessed: 2018-06-20.
[22] Álvaro Rebuge and Diogo R. Ferreira. 2012. Business process analysis in healthcare environments: A methodology based on process mining. *Inf. Syst.* 37, 2 (2012), 99–116.
[23] Manfred Reichert and Barbara Weber. 2012. *Enabling flexibility in process-aware information systems: challenges, methods, technologies.* Springer Science & Business Media.
[24] Roman Vaculín et al. 2011. Declarative business artifact centric modeling of decision and knowledge intensive business processes. *Proc. of the IEEE International Enterprise Distributed Object Computing Workshop, EDOC* (2011), 151–160.
[25] Wil MP van Der Aalst, Maja Pesic, and Helen Schonenberg. 2009. Declarative workflows: Balancing between flexibility and support. *Computer Science-Research and Development* 23, 2 (2009), 99–113.

[1]The opinions expressed in this work do not necessarily reflect those of the funding agencies.