FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Information Systems

Guideline for combining and differentiating between CMMN, DMN and BPMN: An indicator-based use case study

Dominik Gerbershagen

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Information Systems

Guideline for combining and differentiating between CMMN, DMN and BPMN: An indicator-based use case study

Richtlinie zur Kombination und Differenzierung zwischen CMMN, DMN und BPMN: Eine indikatorbasierte Falluntersuchung

Author:	Dominik Gerbershagen
Supervisor:	Prof. Dr. rer. pol. Hans-Arno Jacobsen
Advisor:	Martin Jergler, M.Sc.

Date:	November 15, 2016

Ich versichere, dass ich diese Bachelor-Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, November 15, 2016 Dominik Gerbershagen

Abstract

Acknowledgements

Acronyms

CMMN	Case Management and Notation Model
DMN	Decision Model and Notation
OMG	Object Management Group
BPMN	Business Process Model and Notation
BPM	Business Process Management
ARIS	Architecture of Integrated Information Systems
BPR	Business Process Redesign
IBM	International Business Machines Corporation
BPMI	Business Process Management Initiative
BPEL	Business Process Execution Language
UML	Unified Modeling Language
DRD	Decision Requirement Diagram
CM	Case Management
ACM	Adaptive Case Management
DRG	Decision Requirement Graph
FEEL	Friendly Enough Expression Language
SQL	Structured Query Language
PMML	Predictive Model Markup Language

Contents

List of Figures

List of Tables

Part I.

Introduction

1. Introduction

1.1. Motivation

1.2. Problem Statement

The Business Model and Notation (BPMN) modeling language and technique has become a quasi-standard for modeling business processes, logical steps in software-systems or align companies along the process chains. BPMN was standardized in 2005 by the Objective Management Group (OMG), but had a long list of predecessors including the *Event-driven Process Chain* (EPC), the *Swimlane Visualization*, *Business Process Re-engineering* to mention only a few. Despite the inheritance of these languages, BPMN has some major downsizes which will be discussed. BPMN suits best when it comes to processes that incorporate the *Value Chain Model* by Michael E. Porter [1]. Every company needs these strict processes to optimize the value chain, for the value creation and to separate the hierarchies between employees and departments.
By 2016 these processes have been largely automated. For this automation, the BPMN syntax suited very well due to the strict processes that could be automated more or less easily. The next challenge is to support processes that cannot be simply automated: decisions and case management. At this point, *Case Management Model and Notation* (CMMN) and *Decision Model and Notation* (DMN) come into play. Both are were highly anticipated by people working in modeling departments. This thesis will investigate the benefits of separating the decision logic and case management into these new standards as well as combining them into a macro model. Additionally, every language has its own indicators and way of modeling. The weaknesses and strengthens of BPMN, CMMN and DMN will be presented in detail in order form indicators for when to use which language best. Both modeling languages are relatively new to the market as they were standardized in 2014 (CMMN) and 2015 (DMN). The goal of this thesis is to provide a guideline helping modelers to differentiate between the languages, to help combining each language and to provide information about the historical background

of these new standards. Another key aspect in this thesis is the use case study to prove the indicators by real process models taken from the eKulturPortal GmbH. An evaluation will show how the indicators applied on a model work and how they can work for the reader's own models.

Summing up the preceding paragraph leads to three research questions:

- Investigation of the new Decision Model and Notation specification published by the Object Management Group, extracting the downsides and advantages especially concerning the vast modeling of gateways in BPMN. Is DMN the solution to simplify decision-modeling?
- Investigation of the Case Management Model and Notation specification by the Object Management Group, particularly how case modeling can be applied in a model-driven software development project.
- How do CMMN, DMN and BPMN work together in a model-driven software project? Is there a valid possibility to combine all three specifications in one model? Is it possible to improve the process and information flow, readability and eventually implementation of the model by the developer?

1.3. Approach

To start the analysis of each modeling language, a brief background information about the demand for standardization by people working in modeling departments or researchers will be provided. On this account a thorough literature research suits best, presenting the results in the Background chapter.

In chapter 2 we will examine the specifications provided by the OMG and have a look at each modeling language. What are the strengthens and weaknesses, what has been standardized and when do the standards suit best. These questions will be answered for each CMMN and DMN. BPMN will not be analyzed in detail, but is a key element to do the comparison between the standards. The goal is to carve out the scope, requirements which were met and indicators of each language. In the following chapter, the ability to combine each standard with BPMN will be presented. At first, the modularity of each standard will be determined. Is each language able to stand for its own or does it need to be implemented in a larger model to work properly? Afterwards, the results are taken in order to compare the combination aspect of BPMN, DMN and CMMN. The questions are directed on how to combine each language, whether a model with all three languages is possible or if there are any downsides that prohibit combining the languages. Additionally, the connectors of each model will be carved out.

With these findings obtained we will do the use case study in order to demonstrate the robustness of the indicators. Process models from the eKulturPortal GmbH will be analyzed with regard to the indicators and remodeled to show the working concept of each standard. A demonstration of the connectors, the scope of each language and the requirements which were met will be presented in this chapter.

Closing up this thesis a chapter of evaluation and discussion is succeeding and the conclusion with a summary of the findings and proposals for future work.

1.4. Contributions

1.5. Organization

Part II.

Related Work

2. Related Work

2.1. BPMN

– todo
Geplant: Am Ende der Ausarbeitung

2.2. CMMN

Case Management and knowledge work are not brand new inventions that have been created in the past few years. "Peter F. Drucker made the first reference to knowledge work in (...) 1959 (...)" [2]. A current "overview and research challenges" provide [2] who explain the difference between business process management and adaptive case management. They briefly sum up the state of the art in case management technology and the next generation solutions. Mentioning technology and tools for Case Management, CMMN and Adaptive Case Management, there are many articles dealing with these topics. [3] describe how adaptive case management can be implemented in businesses and integrated in Enterprise Resource Planing systems (ERP). Additionally they approach a new architecture which decouples decision logic, knowledge work and process flows. All this leads to a better handling of information and an optimization of business modeling. Another practical example provide [4] explaining the company's approach towards an implementation of the CMMN paradigm. This includes the ability to change requirements or orders during run-time, which is one of the major aspects in their system. To achieve this goal, they first set up a meta model of their order-based system and enhanced it afterwards. These practical examples are important in order to evaluate the compatibility with the CMMN specification and other modeling languages, specifically BPMN. They also provide a good overview of how to combine modeling techniques and how they are realized as a system in companies. A more theoretical approach to case management and CMMN particularly provide [5] and [6]. They both do research on transforming CMMN into different

languages. [5] do model-to-model transformation from CMMN to DDML (DEVS-driven Modeling Language) which is used to formalize CMMN and analyze it afterwards. [6] have a similar approach, but a different goal. Due to weaknesses of CMMN, the language cannot be used to create a platform for both agile and route processes. They describe agile processes as the ones "(...) of which the exact flow cannot be determined completely a priori" [6], which is a fundamental characteristic of knowledge work and the reason why case management is so important for many industries. Coping with CMMN's downsides they build their platform on a "rule-based cross-perspective and model intermediate language on textual basis, (...) called *Declarative Process Intermediate Language (DPIL)*" [6].
A useful source for evaluating CMMN as a standardization for adaptive case management is [7]. The subtitle *Examining the applicability of a new OMG standard for adaptive case management* is a good foundation to see how OMG met the expectations from the industry and researchers. This paper sets up requirements deriving from different sources described in detail in section two [7]. At the end of their paper, they evaluate how good the requirements were fulfilled by the CMMN standardization and provide feedback for future improvements.

2.3. DMN

The Decision Model and Notation standardization was meant to improve the *separation of concerns* [8] which is the decoupling of decision logic and the control-flow. Biard et al. investigate how the new standard DMN can be used for decoupling BPMN and the decisions modeled as gateways. Decision-modeling is not typically included in control-flow oriented modeling languages. BPMN has not the power to model vast decision-trees due to the gateway restrictions. [9] even calls it a "(...) [misuse] for modeling decision logic". They found an autonomous way of separating the concerns. After averaging more than 900 models from different industries they introduced a "(...) semi-automatic approach to identify decision logic in business processes (...)" [9]. This semi-automatic approach incorporates the 900+ models they used to identify patterns in decision modeling. Formalization is not one of the key issues in this thesis, but the translation of BPMN to DMN or the link between them definitely is. Evaluating the compatibility of DMN with different modeling languages has been the objective of [10]. They approached a combined solution for knowledge-intensive work modeling and extracting the decision logic, what lead them to a new language called *Declare-R-DMN*. Although the Declare language is not part of this thesis, the combination of it with DMN is useful to evaluate the compatibility with BPMN and CMMN.

Part III.

Background

3. Background

3.1. BPMN and its shortcomings

3.1.1. Origins of BPMN

The evolution of business processes and process thinking dates back to the 1980s when Michael Porter developed the *Value Chain* making a first proposal on how to align companies along the business processes [1]. This major step initiated consecutive research and improvements in the field of business process engineering and re-engineering. The latter has been a very popular technique for companies to strip down their legacy processes, optimize them and implement the new ones. A very prominent example is Ford who copied the Mazda's concept of a central database replacing an old fashioned paper stream [11].
Inspired by Ford and Mazda, the *Business Process Redesign* (BPR) was emerging in until the late 1990s. At that time, companies tried to redesign in a radical manner the way processes, people and data keeping works. Dumas et al. define four reasons for the fall of BPR, whereof over-radicalism might be the most influential one [11].

Despite the fall of BPR, the management of business processes became a prominent part of business optimization. Consequential different methodologies like the *BPM lifecycle* or the *Architecture of Integrated Information Systems* (ARIS) were born along with new modeling languages. "ARIS [also] provides a modeling language known as event-driven process chains (EPCs)" [12]. By today, ARIS still is a very popular, thus a well known, methodology for enterprise modeling and EPCs a widely used modeling language for event-driven processes, as the name states.
ARIS was published 1994 by August Wilhelm Scheer, a researcher for Business Information Systems at the Saarland University. Around ten years later, Stephen A. White and the *Business Process Management Initiative*, consisting of roughly 35 individuals and companies total, published the first version of the new *Business Process Modeling and Notation* (BPMN) concept [13]. In 2006, the OMG adopted BPMN as

well as the BPMI and published an overhauled version, BPMN v. 2.0 in 2011. This version was highly anticipated and had to meet high expectations. The goal was to create a language with the ability to interchange modeling tools. New parts were added such as the *Choreography diagram* for modeling data exchange between partners and the *Conversation diagram* showing the relationship between several partners [13].

3.1.2. Shortcomings

"As the name already indicates, BPMN is restricted to process modeling [...]" [12]. The goal in terms of BPMN was to create a powerful tool for modeling traditional processes. BPMN is an interchangeable language predestined as a middleware between the database and software layers. It has the ability to be executed *Business Process Execution Language* (BPEL) and a well-known set of symbols and artifacts for modeling the diagrams. Additionally, it inherited its simplicity from EPCs and the *Unified Modeling Language* (UML), which is also part of the OMG family.
This simplicity, however, has limitations when it comes to agile and decision intensive models. Decisions, for instance, are illustrated by gateways. These gateways work as connectors separating the process flow into different branches. Each branch has a token wandering in flow direction through the graph. But what if a business wants to change decisions in real time without losing the tokens flowing through the chart? At the current state of development, which is BPMN v 2.0.2, there is no possibility to do so. This is only one of the problems which [14] and [15] investigated more closely. They both came to the conclusion that BPMN is lacking the ability implement business rules which leads the user to display them in spreadsheets. Another downside is the impossibility of modeling agile processes. People working in departments which do not have strict processes but more agile ones cannot use the supportive power of BPMN or process optimization. Despite their knowledge intensive work, they still can benefit from optimizing workflows, or even help new employees getting known to their new job in the department. In this thesis, we will not focus on Business Process Model and Notation (BPMN)'s syntax deficits but more on the possibility CMMN and DMN offer.

3.2. Origins of CMMN

Case Management and Notation Model (CMMN) was standardized by the Object Management Group (OMG) in 2014 which represented a peak in terms of notation for the case management research. However, research began more than ten years ago when Wil Van der Aalst first published his paper about case handling offering a new approach for flexible processes [16]. He summarized different opinions from several authors into the statement that contemporary workflow-systems were not able to handle flexible processes. To be precise, he identified four problems:

- work needs to be divided into atomic steps, so called tasks
- the routing of activities was used to distribute the tasks and authorize workers the workers to do them
- *context tunneling*: workers only focused on the process flow but not on the context surrounding the activities
- the individual's focus was more on what *should* be done instead of what *cane* be done

He portrayed a *blind surgeon* as a metaphor who incorporates the four problems in his daily work, for instance refusing a blood sample when it was necessary but not part of the problem in case of what could be done and what should be done [16]. Consequently he solved the problems inventing an adapted model for cases, which consists of objects that can be found in the OMG'S specification from 2014 as well. In 3.1 a comparison of the key objects of Case Handling and CMMN is provided. The Case Handling's key objects are:

- activities or tasks which are the atomic unit of work
- Actors with roles who execute, skip or redo the tasks
- Case representing the product the actor is producing
- data objects for information and values
- forms that present data objects from different point of views

Van der Aalst proved his concept practically in a joint venture with dutch Construction company. They applied the Case Handling model and built a prototype with FLOWer resulting in a rudimentary worklfow management system. They requested workers to use the system during their work and evaluated the prototype in the end. Van der Aalst calculated a Return on Investment (ROI) of 0.7 to 1.4 years in the year 2002 [16]. With today's workflow management systems, the ROI would presumably be much higher.

Table 3.1.: Comparison CMMN and Case Handling Model

	van der Aalst	CMMN Specification
Product	Case	Case
Atomic Unit of work	Activity / Task	Task
Information Handler	Data Object	Case File Items
Information Representation	Forms	
Executing Person	Actors and Roles	Human tasks with asignees
Connectors	not specified	specified
Triggers	not specified	Event Listeners

3.3. Origins of DMN

The Decision Model and Notation language describes processes in a different way as BPMN does. They differ in their way of how to achieve the actual goal of the process. BPMN, for instance, describes an imperative way. Goedertiert et al. define imperative as "[...] [focusing] on providing a precise definition of the control-flow of the business process in a graph-based process modeling language" [17]. In order to model decisions, BPMN has a rich amount of gateways to direct the control-flow into the desired tasks depending on the decisions being made. Additionally BPMN provides a Business Rule Task sending data to and receiving it from a business rule engine [18]. However, this business rule task " [...] cannot be decomposed any further - and involves no user interaction" [19]. Thus the modeler is not able to define roles for decision, define any requirements for these and is not able to decompose them into fine-grained steps. This might be unnecessary for simple decisions, but in larger and more sophisticated diagrams, complex decisions cause huge branches in the control-flow making the diagram confusing the reader. [20] formulated *Seven Process Modeling Guidelines* according to an empirical study. They built an EPC model with various branches connected by *AND*, *OR* and *XOR* connections. Additionally they violated naming conventions and used multiple end points. One of the resulting guidelines (G1) states "Use as few elements in the model as possible" [20], which is contradicting the imperative way of modeling decisions in BPMN. In this situation, declarative modeling approaches are more appropriate. According to Alan Fish, declarative languages "[...] focus on what should be done in order to achieve business goals, without prescribing how an end state should be reached" [19]. The CMMN specification is a declarative modeling approach as well, whereas EPCs and BPMN aren't. Fish also described a way to refine the decision requirements. He invented the Decision Requirement Diagram (DRD) to show " [...] the structure of the required decision-making as a network of decisions and subdecisions with their supporting areas of business knowledge and data" [19]. The main use for this DRD is to make knowledge involved in decision-making explicit instead of implicit. In the Decision Model and Notation (DMN) Specification by the Object Management Group, a DRD can also be found. This DRD includes the elements as Fish invented for his DRD: Decisions, Input Data (Fish: Data area), Knowledge Source (Fish: Knowledge Area). However, the OMG's DRD is a much more refined modeling approach since it incorporates Fish's DRD and is under constant development. Summarizing the findings, the DMN technique is derived from Fish's findings on how to automate decisions and the declarative modeling approaches which supplement BPMN with tools to model processes without strict control-flows.

Part IV.

Indicators

4. Differentiating between BPMN, CMMN and DMN

Ausblick, was bietet dieses Kapitel und was ist das Ziel

Diagramm mit Legende versehen

4.1. BPMN and routine processes

Creating business process diagrams nowadays almost seems to be intuitive. This results from a broad background of routine process modeling incorporating the rise and fall of BPR, ARIS and its EPCs and many more routine process modeling languages. Altogether, they formed the current understanding of business process management and modeling. Since BPMN is a broadly adopted modeling language and a standard that shaped the industry, users might be mislead when to use BPMN correctly. In this chapter, a definition of requirements and indicators will be provided for the correct use case scenario identification in terms of BPMN as well as a short introduction to the model.

Business Process Model and Notation incorporates *Business Process*, which has been defined and redefined many times by today. For this thesis, we use the OMG's definition of business processes:

> Business Processes: business plans include Courses of Action - what the enterprise has to do to achieve its Ends - transformed into Business Processes that ecnompass activities, sequencing, dependencies, interactions, triggering by business events, etc. [21]

Business processes are ways to achieve goals, to align activities along them and to define interactions, decisions and events. This definition basically incorporates all the core elements of BPMN: Activities, paths, swimlanes (for responsibilities), decisions and events. To recap the notation of BPMN, a small example is shown in figure 4.1. In this model, the application process for a student program is shown, which consists of tasks such as *Receive the registration* or *Check deadline*. Tasks can be precised as *Manual tasks*, *Send / Receive Tasks* and have assignees, as shown in *Check*

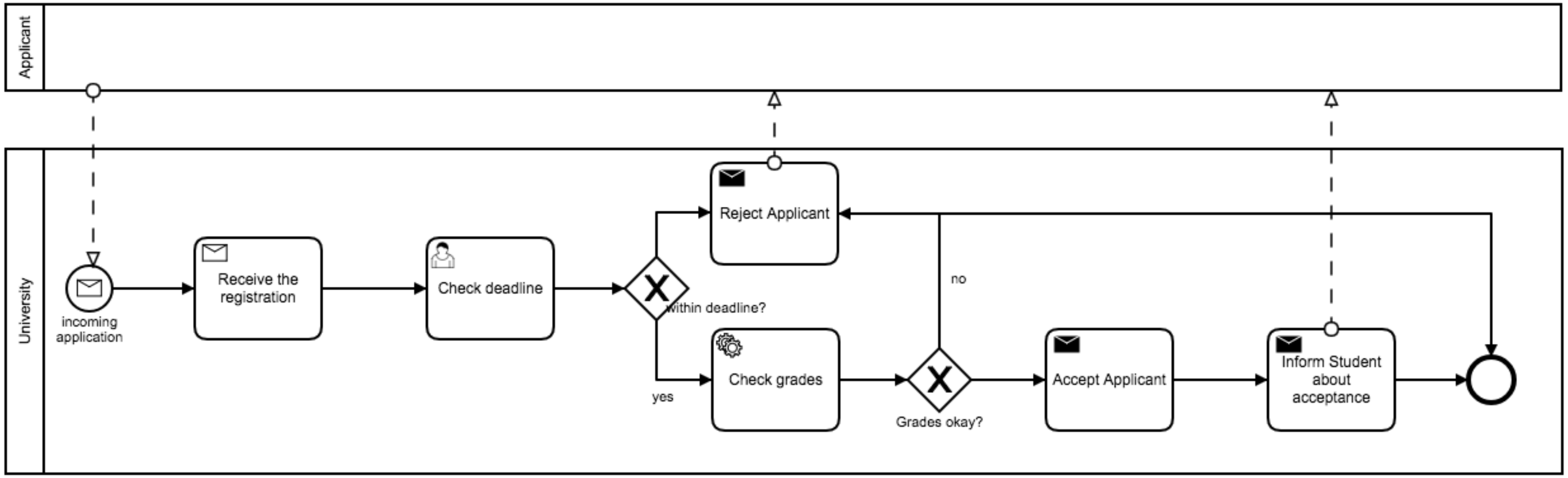


Figure 4.1.: A small BPMN example: the application process for study programs.

deadline. In current Workflowmanagementsystems, this workflow model can be fully automated and the assignee is presented its tasks when they are instantiated, e.g. when an application form was received. BPMN models have a starting and an end point which can also be events. In our example, the starting point is a *Message Event*, that indicates the process has started due to a received message. Decisions are a major concern in every control flow oriented graph and BPMN models usually consist of several decision points. They are notated as diamond-shaped *Gateways*, either empty or with a special symbol inside. Each symbol stands either for *AND*, *OR*, *XOR* or *COMPLEX*. Gateways direct the control flow depending on the decision logic prescribed. In our example, a *XOR* Gateway controls the workflow into either the direction of acceptance or the opposite one, depending on what date the applicant's form was submitted. Additionally, the communication between the workflow's two participants is notated as so called *Pools*, which can be divided into *Swimlanes*. Each pool is a communication partner and a swimlane indicates different responsibilities such as departments. If a pool is not filled with BPMN elements, the communication partner is out of scope of the modeler and the processes cannot be modeled. These pools are *Black boxes*.

As we see in the example, the process is well-structured and has no room for creative decisions or even discussions at run-time. The only way the process can differ, is human intervention or changing the process at design-time. The former is equal to not executing the process in the intended way and the latter is bad for processes in volatile environments. BPMN is suited for a limited variety of processes, foremost business processes. The OMG's specification excludes specifically the following [18]:

- Definition of organizational models and resources,
- Modeling of functional breakdowns,
- Data and information models,
- Modeling of strategy,
- Business rules models

Furthermore, the OMG's specification not only provides exclusions such as the ones mentioned above, but it helps to indicate the correct usage of BPMN. Derived from the background chapter 3 and the example above, the following indicators (table 4.1) serve as a guidance for modelers to identify when to use BPMN correctly.

Indicators	BPMN
General	Predefined sequences of activities with decisions (gateways) to direct the sequence along the alternative paths or for iterations
Type of process	Predefined, fully specified, repeatable, business process, control flow oriented
Type of work	Routine work with possible automation
Type of decisions	Simple, driven by rules or events
Control flow	Strict and necessary
Intervention at run-time	no
Typical application	Value-added chain processes, workflows across companies, departments etc.
Key question before modeling	Does the work consist of routine elements which can be optimized or even automated?

Table 4.1.: BPMN indicators.

Information about processes is often stored in documents, deprecated models or held by knowledge carriers according to Dumas et al. [11]. This information needs to be extracted by the application of the so called *Process Discovery*, which is in short gathering all the information in a technical or manual way before creating the models. Having this information gathered together, the key question before modeling is *Does the work consist of routine elements which can be optimized or even automated?* (see 4.1). Routine processes occur regularly and their flow is "known a priori" according to [6]. Consequently it is not intended to change the process flow during run-time, as routine are well-structured and predefined. This is a major problem for workers whose job is more based on knowledge and their experience than routine work, wherefore is Case Management and CMMN were forged and standardized. In our example, the decisions are simple yes or no questions. One of them, *Check grades*, is a business rule task containing a small table of when a grade acceptable or must be rejected. Overall, the decisions do not require modeling any requirements or personnel apart from the assignee involved in the decision making process. Furthermore, the decisions are always stuck with rules or events that happened before. The process itself is control-flow-oriented and the control flow is necessarily strict since every candidate needs to be processed in a fair and equal manner. In the end, the process could be automated completely with the appropriate workflowmanagementsystem and would not need any assignee to check the deadline or the grades. According to the indicators, the example is modeled using the suitable modeling language as the process was predefined and well-structured. To prove the indicators' validity, a use case example taken from the eKulturPortal

GmbH [1] is provided in chapter .

hier fehlt die Chapter referenzierung!

4.2. CMMN and knowledge work

Routine work occurs in every company, every industry and even in our leisure time. However, not every job is comprised entirely of routine work. If a person is sick and needs to see the doctor, often the doctor's job consists of routine parts and ad hoc parts. Each patient suffers in a different way, has a private or statutory health insurance, is able to understand the doctor's language or not. In each case, the doctor or the medical personnel needs to decide in an ad hoc manner to identify the patient's disease and medicate him accordingly. Overall, this job requires more knowledge and experience than a strict workflow. Applying the BPMN indicators makes the necessity of a different modeling technique obvious as the indicators do not meet the requirements of this example (flexibility instead of a strict workflow, knowledge driven instead of control flow driven). Standardized by the OMG in 2014 [22], the modeling language is intended to meet the requirements of knowledge intensive professions and support workers with models and their execution in workflowmanagementsystems. The above described example diverges from the business processes explained in the BPMN section earlier. Instead of focusing on what *has* to be done in which order, the medical personnel should adhere to what *can* be done, as Van der Aalst et al. described in [23]. This is an essential part of the *Case Management* (CM) and *Adaptive Case Management* (ACM) approach. Case management starts with a case such as a legal case, a patient's disease or ”some other focal point around which actions are taken to achieve an objective” [22]. Each case contains a set of tasks and files necessary to achieve the goal. These sets are ”minimally predefined” [22] resulting in two phases of working with cases: the design phase and the running phase [22]. During the design phase, tasks and information are aggregated in a manner that is very similar to business process designs. As a result, the work is available in a more structured way as it has been before yet with the possibility to perform the work in a completely ad hoc if necessary. Additionally, case management requires the run-time flexibility such as performing tasks depending on the situation, changing the order of tasks or even collaborate with different people involved as intended. At a first glance, this seems counter-intuitive since the common understanding of modeling processes is modeling business processes or similar strictly structured ones. To

[1] see `http://ekulturportal.de/`

oll hier
och eine
egende
ngelegt
erden?

illustrate the paradigm of case management and introduce the CMMN notation, figure 4.2 provides an example . The subsequent explanation is derived from [22] and [24].

The first element that immediately catches the eye in this example is a large folder depicting the `Case Plan Model`, a root element containing all further elements involved with a name tag in the upper right corner. Next, there is a big structure called a stage and illustrated as a rectangle. Similar to BPMN, stages form a group of elements into a sub process or - to put it in the CMMN context - in a sub case. Stages can be connected via sentries and connectors that are the lines between tasks and the diamond shaped exit or entry criteria. Both, exit and entry criteria trigger when events occur. There are three possibilities to make sentries trigger:

- `on [event] if [condition]`
- `on [ event]`
- `if [condition]`

According to the CMMN specification, this provides the opportunity to receive events and check whether they have effect on the following process steps [22]. In figure 4.2 a timing event is called `Deadline missed`. This event is connected to an exit criterion, denoted as a black diamond on the border of the Case Plan Model. When the event is triggered, the sentry checks `on [deadline missed] if [date today > deadline date]` [2] and triggers the exit criterion terminating the case due to the missed deadline. A different event type is the `User Event` listening to user interactions that trigger the exit criterion. Besides these two event types, each task, stage and `Case File Item` can trigger events on their own which are received by sentries. `Case File Items` are containers for pieces of information, for instance the patient's record or in this example the papers which need to be read (not contained in the diagram). They are denoted as files known from the notation as computer files are depicted. Another structural element is the `Milestone`, an oval shaped form representing an ”achievable target” [22] and providing the ability to measure the progress of a case. Each milestone can have multiple entry criteria or none to show that a milestone is reached or not. In general, ”no work is directly associated with a milestone” [22] but rather a ”completion of a set of tasks” or ”the availability of key deliverable” [22]. Figure

[2] Note: This example is just for illustration purpose and incorporates no concrete syntax nor semantics.

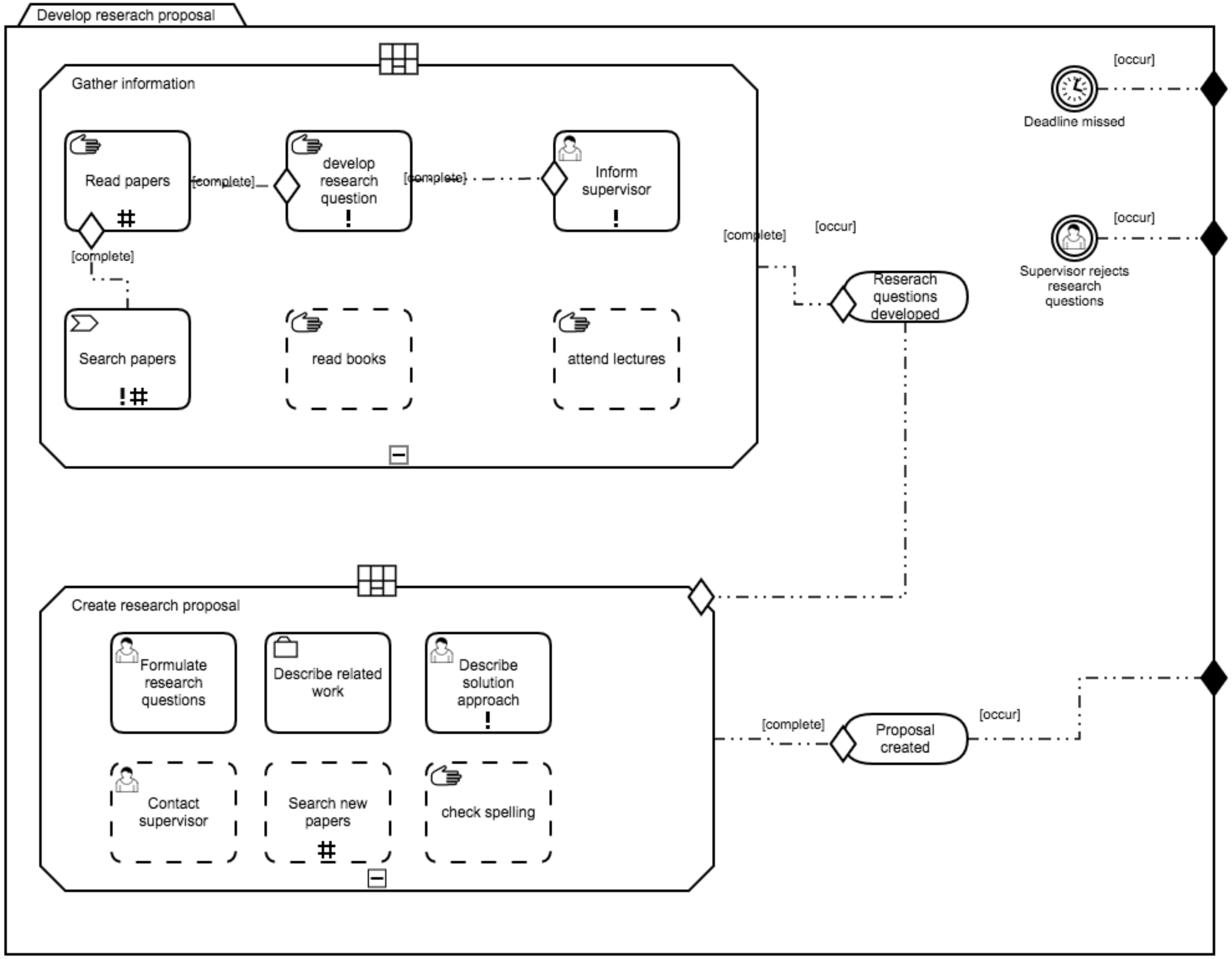


Figure 4.2.: CMMN example: Creating a research proposal.

4.2 contains two example milestones: `Research questions developed` and `Proposal created`. Both milestones are connected to either a stage or an exit criterion indicating a dependency. The first milestone needs to be achieved to create the research proposal, the latter one determines the successful termination of the case.
As mentioned earlier, CM requires run-time flexibility and by now, there has nothing been explained but tasks, events and milestones which are per se inflexible. To guarantee run-time planning, tasks and stages have a discretionary version. Discretionary means, a user may decide during run-time if this task or stage will be executed in a following iteration. At this point, the knowledge and experience comes into play. In our example, the gathered papers might not be enough to provide good citations and emphasize the solution approach. Here, only the writer himself can identify if he needs to do more research to find correct citations as this is a decision requiring knowledge and experience in writing research papers. It is to his or her discretion if the task will be executed or not. Supplementary to `Discretionary Items`, `Planning Tables` indicate the ability to plan `Discretionary Items` and apply control mechanisms called `Applicability Rules`. Case workers are presented `Planning Items` if the corresponding `Applicability Rule` evaluates to true. Overall, this allows the modeler and the case workers to plan at design-time (creating the case model) and at run-time (planning the time discretionary items). The last visible part of the CMMN notation that has not been explained is the type of tasks available. In the example case, there are four types of tasks (each type can be indicated in the upper right corner of the task):

- `Human Tasks` either blocking (indicated with a "User" in the upper right corner) or non-blocking with a hand in the right corner
- `Process Task` with a "Chevron" symbol
- `Case Task` with a folder symbol

Both `Human Tasks` indicate that an assignee with the appropriate role needs to fulfill the modeled work here. Non-blocking signifies this step can be done while other tasks of the model run simultaneously in contrast to the blocking one, which does not allow other tasks to be performed at the same time. A `Process Task` calls a simple business process modeled e.g. in BPMN and a `Case Task` calls a different case potentially independent from the one being executed. In figure 4.2, `Read Papers` is a non-blocking human task, `read books` a non-blocking discretionary human task, `Formulate research questions` a blocking human task,

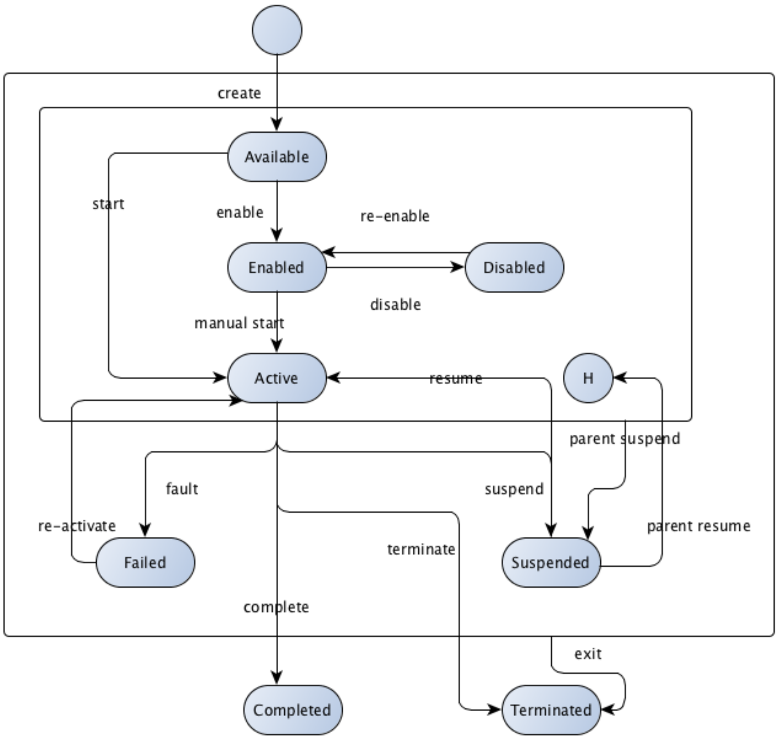


Figure 4.3.: Stage and Task Lifecycle according to figure 7.3 in the CMMN specification.

`Describe related work` a case task and `Search papers` a business process task. Apart from the mentioned types of tasks, there a additional execution semantics as seen in the `Search papers` task. An exclamation mark signifies this task as to perform, a sharp that this task has several iterations and a play symbol indicates the `Manual Activation Rule` (not in the example). This rule is connected to entry criteria and transitions from the status `enabled` to `active` if the criterion's condition is fulfilled. CMMN, in contrast to BPMN, offers a complete lifecycle with corresponding states.

Figure 4.3 is a replica of figure 7.3 in [22] and shows the different states and transitions for stages and tasks. These stages, compared to BPMN, allow the modeler and the worker to measure and to plan work in a different way than business processes are handled. One of the most commonly mentioned downside of BPMN is the lack of states and transitions according to [25]. In the example (fig. 4.2), states are not included since they are not visible in the notation and only occur during run-time. In addition to fig. 4.3, CMMN offers similar states for sentries and the whole case plan model and can be found in the execution semantics chapter in [22].

Indicators	CMMN
General	Workflow can be managed in a flexible way or in a complete ad hoc manner, overall cast cannot be orchestrated by predefined sequence of tasks
Type of process	Depending on evolving circumstances, ad hoc, data centric
Type of work	Ad hoc, not predefined, sparsely routine work
Type of decisions	Stateful conditions, rules, exit and entry criteria
Control flow	Not necessary but can be modeled
Intervention at run-time	Yes
Typical application	Patient care, medical diagnosis, insurance cases, governmental permitting, problem resolution in call centers, sales and operations planning, invoice discrepancy handling, engineering of made-to-order products
Key question before modeling	Does the work comprise unstructured elements and ad hoc workflows requiring knowledge or experience instead of strict rules?

Table 4.2.: CMMN indicators.

Table 4.2 shows a summary derived from the explanation and the example, both planting on the basis of the CMMN specification [22]. The example emphasized that CMMN is in general applicable for not predefined, to a high degree knowledge dependent work. Decisions are modeled in conditions on when to enter or exit stages, task or milestones. Rules such as `Application Rule` or `Repetition Rule` allow the modeler to guide the case worker through tasks and show him or her what tasks have to be done and which have to be done for a specific amount of iterations. However, decisions similar to gateways in BPMN are not in scope of this notation and dedicated to the case worker. In addition, the required flexibility is fulfilled by `Discretionary Items` allowing the workers with the appropriate role to adapt the workflow in during run-time. States as shown in figure 4.3 contribute to this flexibility but also let the involved personnel measure the progress of each case. This results in a more manageable and verifiable way of Case Management.

alic zu
chreib-
naschi-
en Style
endern

orrigieren

4.3. DMN and decision making

In section 4.1 decision-making in BPMN was shown with the aid of a small example, whose excerpt is shown in figure 4.4. Two gateways guide the control flow according to the decisions made. Each gateway compares a value with a given limit and verifies it. If the value is correct, the control flow passes on to the next gateway or, in case the value is out of bounds, rejects the student's application. Figure 4.5 demonstrates a more sophisticated version of the same process. Here, the the university has four requirements but only one needs to be fulfilled to pass the application process and to be accepted for the desired study program. All of the gateways are OR gateways signalizing it doesn't if one of these requirements is not met but one of the other is. At a first glance, figure 4.5 looks more complicated than figure 4.4. Although the actual decision-making process is kept simple, the BPMN notation with its gateways makes it look confusing and, at a much larger scale, for the reader impossible to comprehend the process a whole. At this point, DMN becomes relevant.

The DMN notation comprises two parts: the Decision Requirement Graph with one ore more Decision Requirement Diagrams and the decision tables. Figure 4.6 illustrates the corresponding Decision Requirement Graph (DRG) for the decision-making seen in fig. 4.5. DMN's notation for DRDs contains only four distinct shapes and three different types of connectors. In fig. 4.6, there are three elements provided and one is missing due to the shortness of the example:

- Decision: Decisions are depicted as rectangles and incorporate the decision logic which is mapped in decision tables. In the example there are two decisions, Deadline check and Acceptance rate according to Transcript of Record and both have a table symbol in the upper right corner indicating the decision logic.

- Input Data: Each decision needs inputs and input data is the fundamental and least complex form possible. This element stores information of any type from simple types like dates and integers to lists and functions expressed via Friendly Enough Expression Language (FEEL). An Input Data element may only be connected directly to decisions and knowledge sources.

- Knowledge Source: DMN makes it explicit where knowledge sources are involved in the decision-making. Typically knowledge sources are documents, best practices but also people or bodies of legislation [**?**]. "Knowledge sources are the authorities for a decision [...]" [26], which may be connected

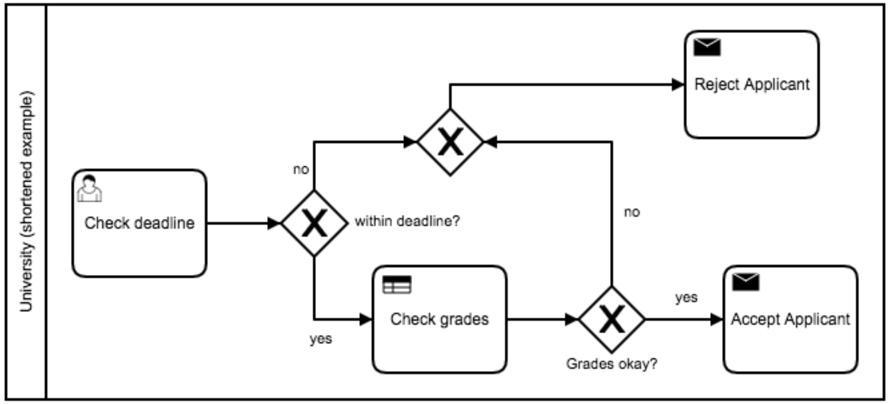


Figure 4.4.: Shortened example from section 4.1 with only two decisions.

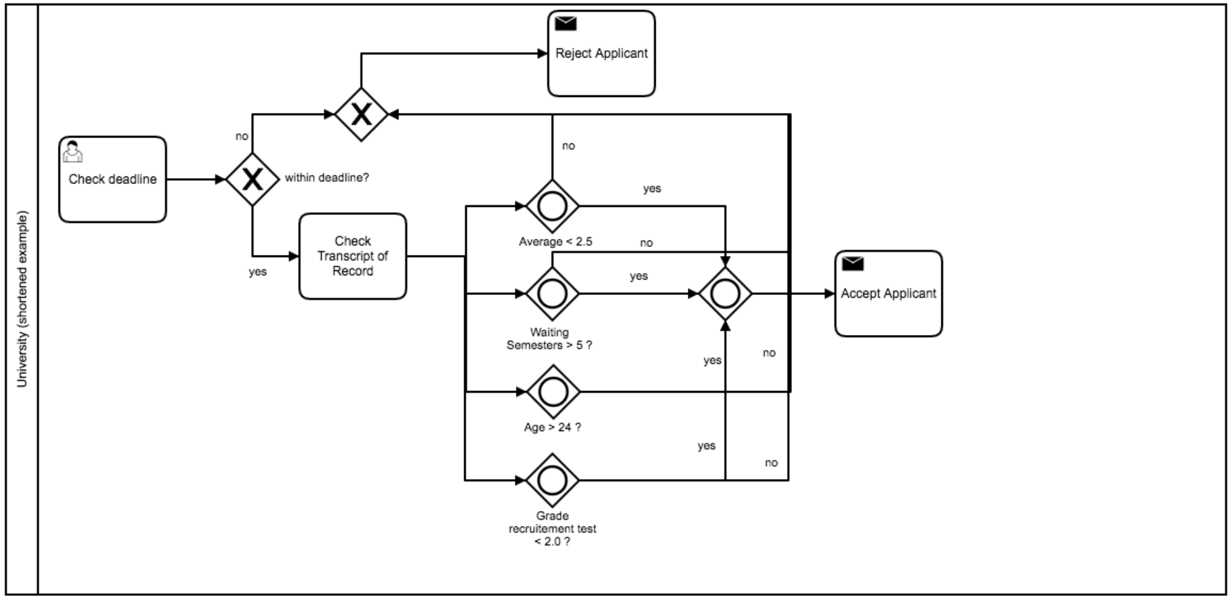


Figure 4.5.: Shortened example from section 4.1 with additional decisions.

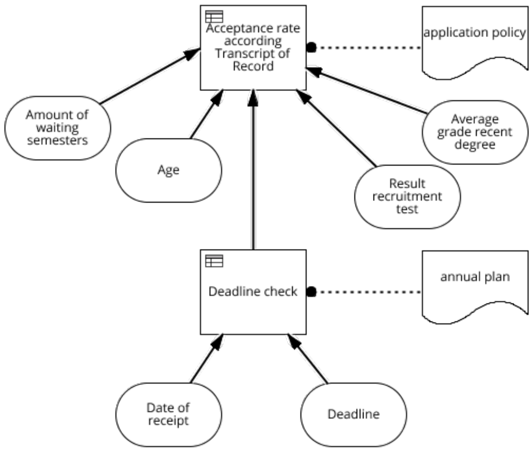

Acceptance rate according Transcript of Record
application policy
Amount of waiting semesters
Age
Average grade recent degree
Result recruitment test
Deadline check
annual plan
Date of receipt
Deadline

Figure 4.6.: A DMN Decision Requirement Graph incorporating the decision-making of the BPMN process in 4.5.

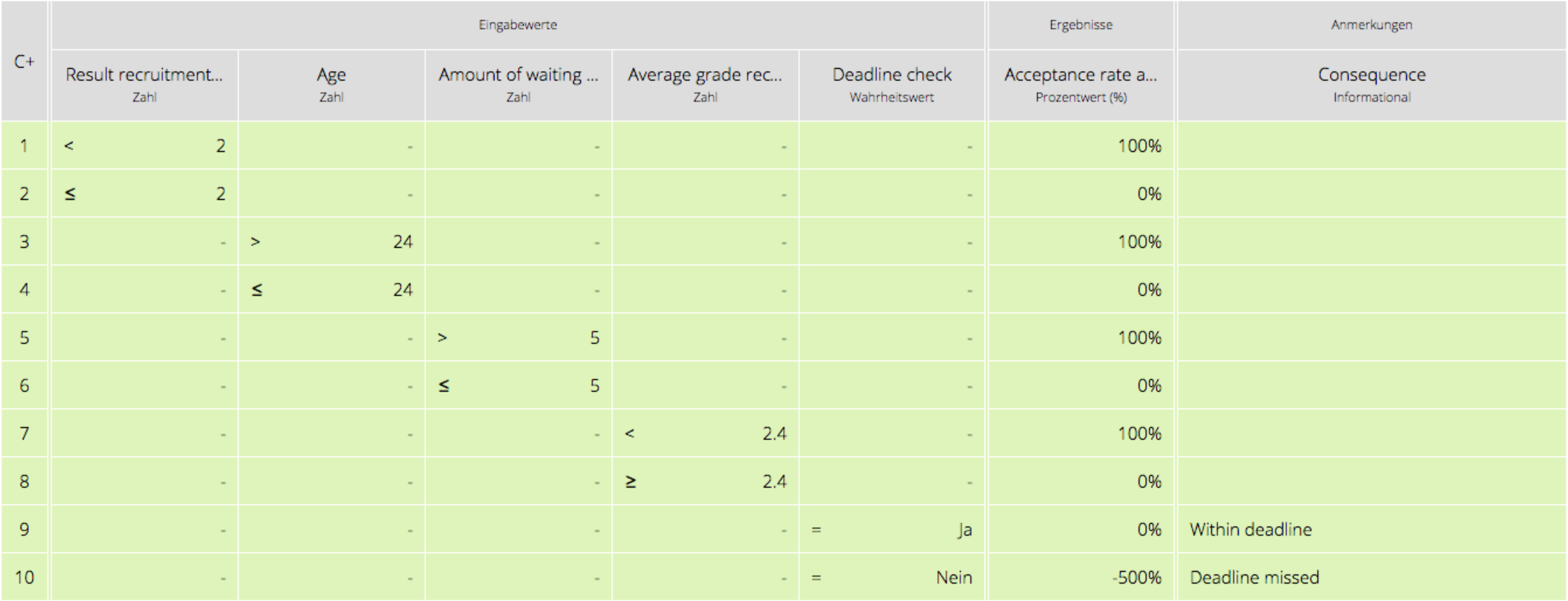
with input data since managers, for instance, might need background information not directly involved in the decision logic. In the example, the application policy and annual plan of the university is used as `knowledge source`.

- `Business Knowledge Model`: In contrast to `Knowledge Sources`,`Business Knowledge Models` are non-authoritative and serve as container for analytic models, decision logic or business rules [?]. It is depicted as a rectangle with cut-off corners in the upper left and lower right. As they only influence decisions directly, the specification allows consequently no connection but with decisions or different business knowledge models.

As stated above, the notation comprises only four elements making it a very compact modeling language in terms of the DRG. In addition, there are three different connectors indicating either an `information requirement` (connections from decisions to decisions or input data to decisions), an `authority requirement` (connections from knowledge models to decisions, business knowledge models or different knowledge sources) or `knowledge requirement` (connections from business knowledge models to decisions or different business knowledge models).

The second part of the DMN specification allows the modeler to create decision tables containing the decision logic. An example table [3] is provided in fig-

[3] Each modeling tool implementing the DMN specification is allowed to vary the orientation of



C+	Eingabewerte					Ergebnisse	Anmerkungen
	Result recruitment... Zahl	Age Zahl	Amount of waiting ... Zahl	Average grade rec... Zahl	Deadline check Wahrheitswert	Acceptance rate a... Prozentwert (%)	Consequence Informational
1	< 2	-	-	-	-	100%	
2	≤ 2	-	-	-	-	0%	
3	-	> 24	-	-	-	100%	
4	-	≤ 24	-	-	-	0%	
5	-	-	> 5	-	-	100%	
6	-	-	≤ 5	-	-	0%	
7	-	-	-	< 2.4	-	100%	
8	-	-	-	≥ 2.4	-	0%	
9	-	-	-	-	= Ja	0%	Within deadline
10	-	-	-	-	= Nein	-500%	Deadline missed

Figure 4.7.: An example DMN decision table with the decision logic accord to the BPMN example in fig. 4.5.

Policy	Variations
Single hit	Unique (no overlap allowed)
	Any (overlap allowed, outputs must be equal)
	Priority (returns matching rule with highest output priority)
	First (first hit by rule order)
Multiple hit	Output order (returns all hits in decreasing output priority order)
	Rule order (returns all hits in rule order)
	Collect (aggregates all hits with specified operators: sum, min, max, count)

Table 4.3.: DMN hit policies cited from the DMN specification [**?**].

ure4.6 implementing the decision logic according to the BPMN example in figure 4.5. Each table comprises of several input columns and at least one output column. An input can be a string, integers, floats or basically any type of numbers required. In this example table, there are five inputs and one output with annotations such as `Deadline missed`. Each row signifies a rule in the following syntax: `If [INPUT A] AND [INPUT B] then [OUTPUT C]` [**?**]. The inputs are logically connected by the boolean `AND`-operator resulting in as many outputs as desired, if a rule matches. For every input in the decision table, there is an `input data` element modeled in the related DRG. This also applies for decisions and subdecisions. Comparing the table with the BPMN example demonstrates a downside of the decision table: the "`AND` only" syntax. However, modeling `OR` connections is possible with a workaround. Besides the table's data and rules, business analysts can also define `hit policies`. These policies are necessary if some of the rules overlap. In the example table (fig. 4.6), all of the rules are intended to overlap making a hit policy necessary. There are two major categories of policies: `single hit` and `multiple hit`. The former one allows only to trigger one rule, the latter one allows several rules to trigger. A brief overview with an explanation can be found in table 4.3.

To achieve an `OR` connection of the rules, the hit policy is set to `Multiple Hit: collect (sum)` and the measuring unit of the output is set to percentage. Consequently, the acceptance rate is measured in percentage. The applicant is able to achieve zero or more than 100 percent according to the `collect: sum` rule. Another problem of this method is the inflexibility of the outputs. In this example,

DMN Tabelle probieren auf Englisch umzustelle

decision tables and the concrete look, such as colors or shades.

an output value of `true` or `false` would have been the best way to model the decision, but neither the decision table nor FEEL allow the aggregation of boolean values. Last of all to mention is the sub-decision `Deadline check`. The decision logic is not implemented in this table, but is referenced by a another decision table (see fig. 4.6 for illustration). A simple check written in FEEL counts the difference between the application's date of receipt and the given deadline. If the result is less then zero, the deadline is missed and the output of `Deadline check` `false`. Otherwise, if the result equals zero or more, the result it `true`. To individualize the decision logic or create more advanced rules, the DMN specification aggregated several functions and types into an expression language called Friendly Enough Expression Language. It is similar to Structured Query Language (SQL) and Predictive Model Markup Language (PMML) making it easier to understand for non-developers. The expression used in the `deadline check` table is also a predefined function with two inputs (deadline date and date of receipt) and a range of numbers implemented implemented as two rules: `DayDiff(Date of receipt, Deadline)`.

In review, the examples illustrated the multiple uses of the DMN language. Decision tables are useful for automating data intensive decisions and replacing vast BPMN gateway trees. In the same way it is possible to model decisions that require rather knowledge, best practices or analyses than a large amount of data by skipping the decision logic part and only define the dependency graph. On the contrary, it is not possible to any sequence with the DMN notation but the dependency of decisions or the amount of iterations. Although the specification emphasizes the independence of the the notation [?], the DRG on its own might have few contexts where a standalone DRG is the most convenient solution.

hecken,
b eine
ewer-
ung hier
rlaubt ist

Table 4.4 summarizes the knowledge acquired by analyzing the DMN specification [?], Debevoise and Taylor [26] and [?], as well as creating the provided examples in this section. It was shown that DMN is an applicable solution for processes that focus on decisions or require decisions, thus are complex to illustrate in BPMN. The most common applications are calculations of salaries, provided in [?] or making a choice by several influencing factors, for instance choosing the right vehicle for hazardous material [26]. Particularly the breakdown into two parts, the Decision Requirement Graph and the decision tables, makes it simple for the modeler to decide between decisions that only need data, therefore use the decision table, model the dependencies of decisions and the decision-makers, ergo utilize the DRG, or combine both.

isschen
esser
us-
uehren

Indicators	DMN
General	Detailed decision modeling and adjusting during run-time, Decision Requirements Graph and Decision Tables incorporating decision logic and involved authorities / dependencies
Type of process	decisions only, but with all the requirements and input data
Type of work	human decision-making, automated decision making, aggregation of knowledge, preparation of decisions
Type of decisions	complex, data-centric, many requirements, engagement of human decision-makers
Control flow	dependencies only
Intervention at run-time	yes
Typical application	calculation of discount rates, salary, multi-instance decisions, cross divisional decisions involving several deciders, aggregation of data to get a specific output for a specific case, inter-divisional decision-making
Key question before modeling	Are there decisions that need to be adjusted during run-time, are data-intensive or require the modeling of extra information (policies, analytical models) or roles (responsible managers)?

Table 4.4.: Indicators for the identification of DMN use cases.

4.4. Summary

In this chapter, each specification was presented with examples and their characteristics. By analyzing the OMG's standards and further research publications, these characteristics could be transformed into indicators helping modelers to identify when to use the corresponding technique. Additionally, the examples explained the fundamental principles of the notations and how to use them correctly. The main focus, however, was on developing the indicators. For further information about the notations and more sophisticated examples or instructions, please refer to the corresponding OMG specification or the provided sources in the references chapter. Concluding the preceding sections, BPMN is useful for predefined processes, mainly mapping routine work without the possibility for run-time intervention; CMMN is beneficial for people working in volatile environments with routine parts but mainly work depending on their personal experience and knowledge; and DMN is a valuable way to encapsulate complex decision-making in a separate, compact notation that allows run-time adjusting. For all of the mentioned notations, there are overlaps that make it possible to combine them and creating a whole model with separate notations for each domains. This will be

iese ue-
erschei-
ungen
er spec-
ications
uss erst
och ver-
iziert
erden!

discussed more detailed in the following chapter.

Todo list

Appendix

Bibliography

[1] Micheal Porter Michael E. Porter. *Competitive Advantage: Creating and Sustaining Superior Performance*. FREE PR, 1998.

[2] H. R. Motahari-Nezhad and K. D. Swenson. Adaptive Case Management: Overview and Research Challenges. In *2013 IEEE 15th Conference on Business Informatics*, pages 264–269, July 2013.

[3] Łukasz Osuszek and Stanisław Stanek. Knowledge Management and Decision Support in Adaptive Case Management Platforms. In *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*. Polish Information Processing Society PTI, October 2015.

[4] Vadim Kuzin and Galina Kuzina. *On the Move to Meaningful Internet Systems: OTM 2013 Workshops: Confederated International Workshops: OTM Academy, OTM Industry Case Studies Program, ACM, EI2N, ISDE, META4eS, ORM, SeDeS, SINCOM, SMS, and SOMOCO 2013, Graz, Austria, September 9 - 13, 2013, Proceedings*, chapter CMMN Implementation in Executable Model of Business Process at Order-Based Manufacturing Enterprise, pages 112–123. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[5] Shaowei Wang and Mamadou Kaba Traoré. A System-theoretic Approach to Case Management. In *Proceedings of the 2014 SpringSim Poster Session*, Posters '14, pages 31–32, San Diego, CA, USA, 2014. Society for Computer Simulation International.

[6] Michael Zeising, Stefan Schönig, and Stefan Jablonski. Towards a Common Platform for the Support of Routine and Agile Business Processes. In *Proceedings of the 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 94–103. Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (ICST), October 2014.

[7] Matthias Kurz, Werner Schmidt, Albert Fleischmann, and Matthias Lederer. Leveraging CMMN for ACM. In *Proceedings of the 7th International Conference on Subject-Oriented Business Process Management - S-BPM ONE '15*. Association for Computing Machinery (ACM), 2015.

[8] Thierry Biard, Alexandre Mauff, Michel Bigand, and Jean-Pierre Bourey. *Risks and Resilience of Collaborative Networks: 16th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2015, Albi, France,, October 5-7, 2015, Proceedings*, chapter Separation of Decision Modeling from Business Process Modeling Using New "Decision Model and Notation" (DMN) for Automating Operational Decision-Making, pages 489–496. Springer International Publishing, Cham, 2015.

[9] Kimon Batoulis, Andreas Meyer, Ekaterina Bazhenova, Gero Decker, and Mathias Weske. Extracting Decision Logic from Process Models. In *Advanced Information Systems Engineering*, pages 349–366. Springer Science Business Media, 2015.

[10] Steven Mertens, Frederik Gailly, and Geert Poels. Enhancing Declarative Process Models with DMN Decision Logic. In *Enterprise, Business-Process and Information Systems Modeling*, pages 151–165. Springer Science Business Media, 2015.

[11] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo Reijers. *Fundamentals of Business Process Management*. Springer Science & Business Media, Berlin Heidelberg, 2013.

[12] Marc Lankhorst. *Enterprise Architecture at Work*. Springer Science Business Media, 2009.

[13] T. Allweyer. *BPMN 2.0: Introduction to the Standard for Business Process Modeling*. Books on Demand, 2010.

[14] V. Repa and O. Zelenzfik. Methodological limitations of modeling languages BPMN and ARIS. In *Proc. 15th Int Control Conf. (ICCC) Carpathian*, pages 507–512, May 2014.

[15] Jan C. Recker, Marta Indulska, Michael Rosemann, and Peter Green. How Good is BPMN Really? Insights from Theory and Practice. In Jan Ljungberg and Magnus Andersson, editors, *14th European Conference on Information Systems*, Goeteborg, Sweden, 2006.

[16] W.M.P. van der Aalst, M. Stoffele, and J.W.F. Wamelink. Case handling in construction. *Automation in Construction*, 12(3):303–320, may 2003.

[17] Stijn Goedertier, Jan Vanthienen, and Filip Caron. Declarative business process modelling: principles and modelling languages. *Enterprise Information Systems*, 9(2):161–185, sep 2013.

[18] Business Process Model and Notation. Technical report, The Object Management Group, 2013. Version 2.0.2.

[19] Alan N. Fish. *Knowledge Automation: How to Implement Decision Management in Business Processes*. JOHN WILEY & SONS INC, 2012.

[20] J. Mendling, H.A. Reijers, and W.M.P. van der Aalst. Seven process modeling guidelines (7PMG). *Information and Software Technology*, 52(2):127–136, feb 2010.

[21] Business Motivation Model. Technical report, The Object Management Group, 2015.

[22] The Object Management Group. *Case Management Model and Notation version 1.0*. Object Management Group, version 1.0 edition, May 2014.

[23] Wil M.P. van der Aalst, Mathias Weske, and Dolf GrÃ¼nbauer. Case handling: a new paradigm for business process support. *Data & Knowledge Engineering*, 53(2):129–162, may 2005.

[24] Knut Hinkelmann. Case Management Model and Notation - CMMN, 2015.

[25] Jan Recker. Opportunities and constraints: the current struggle with BPMN. *Business Process Mgmt Journal*, 16(1):181–201, feb 2010.

[26] Tom Debevoise, James Taylor, Jim Sinur, and Rick Geneva. *The MicroGuide to Process and Decision Modeling in BPMN/DMN: Building More Effective Processes by Integrating Process Modeling with Decision Modeling*. CreateSpace Independent Publishing Platform, USA, 2014.