UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Fodio Tureta Abubakar

A Systematic Literature Review on Business Rule Modeling

Master's Thesis (30 ECTS)

Supervisors:
Fabrizio Maria Maggi
Fredrik Payman Milani
Marcelo Sarini

Tartu 2019

Table of content

A Systematic Literature Review on Business Rule Modeling

Abstract Business processes are constantly evolving and the ways in which processes are managed are also evolving. To keep track of, modify, and manage business processes, business owners are looking for ways to organize their business processes in a way that they are easy to manage. Beyond that, there is also the need to not just manage those processes but also to manage the rules upon which those processes are executed. The business rules that govern an organization's policies are, in most cases, written as documents within the organization's files. To make it easier to choose alternative languages to model those rules that regulate business processes, this paper presents a Systematic Literature Review (SLR) on how business rules can be modeled. We looked at the different notations that can be used to model business rules. We also looked at how they can be incorporated within the business processes. We then provide a list of tools that help to model, verify, and execute business process models enhanced with business rules. To identify these characteristics, we conducted an SLR and built a framework for classifying existing notations to model business rules.

Keywords: Business rules, Notation, Declarative Models, Business process, SLR, Language, Tool Support

CERCS: P170

Süstemaatiline kirjandusülevaade ärireeglite modelleerimise kohta

Lühikokkuvõte Äriprotsessid arenevad pidevalt ja arenevad ka protsesside juhtimise viisid. Äriprotsesside jälgimiseks, muutmiseks ja haldamiseks otsivad ettevõtete omanikud võimalusi oma äriprotsesside hõlpsaks haldamiseks korraldamiseks. Peale selle on vaja mitte ainult hallata neid protsesse, vaid ka hallata reegleid, mille alusel neid protsesse täidetakse. Ettevõtte reegleid reguleerivad ärieeskirjad kirjutatakse enamasti organisatsiooni toimikutes olevate dokumentidena. Äriprotsesse reguleerivate reeglite modelleerimiseks on alternatiivsete keelte valimise hõlpsamaks muutmiseks käesolevas dokumendis esitatud süsteemne kirjanduse ülevaade, kuidas ärireegleid saab modelleerida.

Vaatasime erinevaid märkeid, mida saab kasutada ärireeglite modelleerimiseks. Vaatasime ka, kuidas neid saaks äriprotsessidesse kaasata. Seejärel pakume loetelu tööriistadest, mis aitavad modelleerida, kontrollida ja käivitada ärireeglitega täiustatud äriprotsessimudeleid. Nende tunnuste tuvastamiseks viisime läbi peegelkaamera ja rajasime raamistiku olemasolevate märkuste klassifitseerimiseks ärireeglite modelleerimiseks.

Võtmesõnad: Ärireeglid, Märge, Deklaratiivsed mudelid, Äriprotsess, SLR, Keel, Tööriista tugi

CERCS: P170

1. Introduction

Business processes play a very important role in the smooth development of the business of an organization. They serve as the lifeline of the business and help the business to optimize the use of resources and streamline activities. With a properly specified business process, it is easy to identify which tasks are important in order to achieve the business goals of an organization. Task standardization, avoidance of noise, and improved efficiency are also achieved through properly planned business processes. Business processes form the backbone of any organization and help the smooth execution of the organization's activities and ultimately help the organization to achieve its goals.

For properly executing business processes, there have to exist guidelines and principles governing how the business is run and what the business seeks to achieve. A business rule, according to the Business Rules Group [11], is "*a statement that defines or constrains some aspect of the business*". Some of these rules are written in the form of an organizational handbook. Others could be informal [10]. Some of the rules could be enforced automatically and others require human intervention. Business rules can either be modeled as part of a business process or collected together into a structure called a decision model.

The goal of business rules is to uphold the structure and behavior of an organization. All organization guidelines and statements that govern how the business is run collectively form the business rules of the organization. For instance, a bank may have a constraint that requires a customer to have an account with the bank for a certain period of time with a certain number of transactions before they can accept a loan application from the customer. This business rule or constraint governs the activity *accept loan application*. The following are some examples of business rules:

- Delivery should be made after payment confirmation.
- If payment has not been received after 14 days, the order must be moved to debt collection.
- Credit should only be granted if the applicant is employed.

A business rule is usually expressed in simple, unambiguous terms as a constraint or a condition. The condition could result from an event like the expiry of the payment period, an action being completed like delivery, or a requirement being fulfilled.

Business rules modeling provides an instrument to capture the business structure, strategy, infrastructure, and all other aspects of a business process. This helps not only in easing the execution of the processes, but also makes it easy to track, modify, or improve them. Several modeling tools are available for defining business rules. Some of them such as the Declare tool can also be used for enforcing the rules during process execution. There are also tools that can be used for analyzing business processes based on business rules. For example, ProM [59] can be used to discover business rules from execution data. This helps the business analyst to automate the discovery of rules from their logs. In addition, there are methods to check the conformance of process executions recorded in event logs against a set of business rules [30, 144]. For each modeling language identified in this thesis, we provide a list of tools supporting that specific notation.

1.1 Problem statement

Business decisions play a vital role in the success or failure of an organization. How these decisions are structured helps the organization to achieve its goals. There has been a lot of research in the area of business rule modeling. Declarative process modeling is slowly gaining momentum, as it represents a rule-based approach to business process modeling.

There have been studies carried out on many business rule modeling notations as well as other notations that deal with the modeling of the organization's decisions. However, to the best of our knowledge, there has not been any research that focuses on studying the available business rule modeling notations so as to give a comparative overview of the existing notations. Research in this area is of great importance as it will help organizations to decide which framework or modeling language to use based on criteria such as flexibility, tool support, expressiveness, level of automation. To this end, this paper presents a comprehensive Systematic Literature Review (SLR) on how business rules are modeled analyzing different methods, notations, and semantics that have been used to capture business rules. The papers reviewed are used to build a framework for classifying existing approaches.

1.2 Research question

The main objective of an SLR is to identify, select, and critically appraise an unbiased, comprehensive list of relevant papers in order to answer clearly formulated questions. A literature review is systematic if it is transparent, objective, and replicable. It also has to be based on clearly formulated questions, employ a rigorous and well-defined approach to reviewing the literature in a specific subject area, and summarize the evidence by using an explicit methodology.

We defined a research question (RQ) that helped us to perform an SLR through which we generated a framework for classifying existing methods to model business rules:

RQ: *What criteria can be used to define a framework that describes how business rules can be modeled?*

This research question is vital in tackling the problem statement. By answering this question, we will be able to identify the different notations that can be used to model decisions and business rules and to identify the different aspects of the notations that are relevant when choosing a modeling language. This, we believe, will help organizations to identify the notation that is best suited for their business models in order to improve efficiency.

The rest of the thesis is structured as follows. In Section 2, we introduce the main concepts that will be used throughout the thesis. In Section 3, we present the research methodology. In Section 4, we discuss the result of the research. Section 5 introduces the proposed framework, while Section 6 concludes the thesis and spells out directions for future work.

2. Background

This section introduces some of the concepts that will be used throughout the thesis. These concepts form the core components upon which this thesis is structured. Understanding these concepts will give the reader the necessary instruments to understand the thesis.

2.1 Business Process Management

A business process specifies the activities that are carried out within an organization by employees or equipment, which yields a product or a service for the consumption of its customers. In order words, the business process of an organization is what the organization does to serve its customers. These processes can be either modeled using the available modeling tools and notations or written in a document that guides the employees on how the processes are to be executed.

Business Process Management (BPM) covers all the aspects of managing a business process and how an organization creates, redesigns, and analyzes the processes that make up the core of the organization's business [203]. The BPM lifecycle involves a series of steps through which the business process goes in order to achieve the optimal process. These steps include design, modeling, execution, monitoring, and optimization [203]. The design step involves data collection and the design of the "to-be" process including process flow and operating procedure. The modeling stage involves the theoretical design of the process, the variables with which the process will operate, and a what-if analysis of the process. After modeling, the process is then executed. The execution stage can either be done manually by humans or automated with the help of a machine. This helps to visualize how the process will run, check for conformance and consistency, and also determine ways to improve the modeled processes. The monitoring stage helps to keep track of individual processes and to determine the performance of each task in the process and of the whole process in general. Through this, the problems that might occur in the execution of a process, or of a process task can be identified and corrected. The performance statistics coming from the monitoring stage are used to optimize the process in the optimization phase [203]. The problems identified can be tackled along with other enhancements that can be applied to the process models thus starting a new iteration of the lifecycle.

2.2 Business process modeling

Business processes can be discovered and modeled using other models, text descriptions specified in natural language, or through automated process discovery techniques [133]. Automated process discovery techniques are a set of techniques involving the use of a tool to automatically discover processes from execution logs. This process, for the most part, does not require human intervention [133]. The most widely used process mining tool for the discovery of declarative processes is ProM [59]. There exist tools that are used to generate business processes from natural language specifications [97]. The discovered processes are modeled using one or more modeling notations.

Notations are the modeling languages or tools that are used for modeling business processes, business rules, or decisions in an organization. These notations provide either a graphical set of symbols that are used to capture different aspects of a business [26] or a structured form of natural language [87]. The graphical icons have a different execution semantic that represents how they interact with the execution engine in order to run the process. Business Process Modeling Notation (BPMN) is an example of notation that is used for capturing business processes. Business rules can be captured using a notation such as Fuzzy Control Language (FCL) [169]. Declare [40] captures the business process as well as business rules. Terms notation and language are used interchangeably in this thesis. There are different approaches to modeling business processes and business rules. These approaches or paradigms are discussed below.

2.2.1 Process modeling paradigms

Business processes can be modeled using different approaches. The two approaches that have gained the interest of researchers and business experts are the declarative and the imperative modeling paradigms.

2.2.1.1 Declarative process modeling

Declarative process modeling is an approach to business process modeling where business rules are explicitly stated and used to constraint the behavior of the business process [15, 26,40]. In other words, this modeling approach specifies what rules are not to be violated as opposed to explicitly state how the process should be executed [26]. This approach gives the process participants the flexibility of executing the process in any way that does not violate the rules. Notations such as Declare and DCR Graphs follow this approach.

2.2.1.1 Imperative process modeling

Imperative process modeling, on the other hand, follows a strict procedural specification of a business process where the focus is on the "how" and not on the "what" [26]. This approach requires the modeler to represent exactly how the business process should be executed. In this approach, business rules are often not explicitly represented in the model. They are either represented in a different notation that is taken into consideration during execution in parallel to the main process model or represented implicitly within the process model. BPMN and Petri Nets [151] models are imperative in nature. A disadvantage of this approach is that it restricts the execution of the model. While in the case of declarative modeling, the execution can be performed in any manner as long as it does not violate the constraint, imperative modeling restricts the execution to exactly what has been specified within the model [26]. On the other hand, imperative models provide support, which is vital in the context of pre-defined, predictable procedures.

2.3 Business rules

Business rules specify how the processes defined above are to be executed. They constrain how the activities should be executed within the business process [26]. The policies upon which the organization's processes are carried out are the organization's business rules. Business rules can be textually written or modeled [10]. They can be modeled as part of a business process model using the declarative approach with notations such as Declare [26] or as a separate model that the business process takes into consideration during execution in order to execute its tasks without violations [157]. Whether they are modeled as part of a process model or separately, business rules are the guidelines upon which a business process is executed.

Business rules can be modeled in different ways, e.g., as a simple decision table, in controlled natural language, or as a high-level visual representation [156]. According to [156], business rules can be of different types: reactive rules, which are in the form of event-condition-action, integrity rules for consistency maintenance, and derivation rules. [156] further classifies derivation rules into facts, implicational inference, and queries.

3. Systematic Literature Review

The aim of this study is to provide a framework for classifying how business rules are modeled. Using an SLR, we identify the different methods, semantics and notations that are used to capture business rules. Using this, we build a framework for classifying how business rules can be modeled. The SLR approach is used here for a number of reasons. First, it allows the summarization of all the available literature that deal with the topic [1]. The SLR approach also makes it possible to identify research gaps leading to other areas that could be considered for future research related to the current research. Since this research is concerned with generating a framework, SLR is the best mechanism to identify the proposed framework [1,6].

3.1 Planning of the Systematic Literature Review

A lot of research has been carried out in order to come up with or improve modeling tools for capturing business rules. The SLR will help us to sieve through these research papers in order to find the studies that best help us to answer the research question and build the classification framework. The SLR also helps to summarize the evidence extracted from the papers being reviewed [1]. In addition to finding what has been researched in the field, SLR also helps to discover gaps in the research area [3].

In accordance with the guidelines proposed by Kitchenham [1], this SLR is composed of three stages. The first stage is planning which includes the motivation of the review, outlining the research questions, and the development of the review protocol. The second stage consists of conducting the literature review. This is where we identify relevant studies, select primary studies, and carry out quality assessment [3,4]. It also includes data extraction and data synthesis. In the final stage, we bring everything together into a well-structured, formatted report. In this section, we focus on the first stage of the SLR.

The research question introduced in section 1.2 can be decomposed into six sub-questions. This helps us to extract more detailed information and gives a clearer picture of the characteristics of existing business rule modeling notations. The sub-questions we consider are:

RQ 1.1: *What notations exist that can be used to model business rules and decisions?*

RQ 1.2: *What perspective can the business rule modeling notations capture?*

RQ 1.3: *How are these rules integrated within business process models?*

RQ 1.4: *What kind of tool support exists for the existing modeling notations?*

RQ 1.5: *To what extent do the modeling notations support automation?*

RQ 1.6: *In which scenarios have these notations been applied?*

3.2 Search strategy

To find the relevant body of literature discussing business rules, we used primary and secondary search strategies [2-4] in order not to omit any relevant study that might be of importance to the literature review. The primary search was carried out using several electronic libraries. After thoroughly carrying out the primary search, we proceeded with the secondary search using forward and backward tracing.

3.2.1 Primary search

The aim of the primary search is to gather a comprehensive body of literature relevant to the topic by querying different search sources using different search strings.

a. Search strings

For the development of the search strings, we followed the guidelines suggested by Kitchenham et.al. [1]:

(1) The term "business rules" is key and derived from the scope of the study. Another term used interchangeably with "business rule" is "business constraint".

(2) The term "business process" is also derived from the research questions and is part of the context in which the study is carried out.

(3) The term "notation" is also relevant to the research as it addresses the main issue that the research is trying to solve, i.e., how to model business rules, which is only possible using some modeling notations. It is also a key term in more than half of the research questions.

(5) Another term used in the search, which also relates to business rules modeling is "decision table" as it is also used as a means to structure and manage business rules.

Other terms used in the search, which are either synonymous of the terms used, or are part of the research context, or of the research questions are:

a) automation
b) language
c) declarative process models
d) semantics
e) grammar
f) formalization
g) formal
h) specification
i) syntax
j) definition

The resulting search strings are:

ST1: *(((“business rules” OR “business constraint”) AND “business process”) AND “notation”)*

ST2: *(((“business rules” OR “business constraint”) AND “business process”) AND “automation”)*

ST3: *(((“business rules” OR “business constraint”) AND “business process”) AND “language”)*

ST4: *("Business Rules" OR "Business Constraints" OR "Declarative Process Models") AND ("Semantics" OR "grammar" OR "formalization" OR "formalisation" OR "formal" OR "specification" OR "definition" OR "syntax")*

b. Search sources

The following electronic databases were selected based on their coverage of journal papers and conference proceedings in the field of computer science and BPM:

- Scopus
- Web of Science

These databases provide a comprehensive set of papers that is sufficient for this research context. With over 47 million records, Scopus is one of the largest databases for quality web sources and research literature. Web of Science also has a reputation for being one of the best sources for research literature with over 37 million records dating back to 1945. Both provide a search engine for extracting literature based on different criteria.

3.2.2 Secondary search

The secondary search serves to screen the large body of literature identified during the primary search based on selection criteria.

3.2.2.1 Selection Criteria (Relevance)

Selection criteria help to identify relevant literature that provides the information required to answer the research questions. The selection criteria can be exclusion or inclusion criteria [10]. The letters in braces at the end of each criterion specifies what to do in case of a positive answer to the question. (I) indicates that the paper should not be excluded in case of positive answer and (E) means the paper should be excluded in case of a positive answer. The inclusion criteria we considered are:

IC1: *Is the study within the domain of business rules or business processes? (I)*

IC2: *Does the study introduce a new notation or extends or apply an existing language or notation? (I)*

The aim of the study is to define a framework that describes how to model business rules and decision models. The first inclusion criterion includes any study where business rules or business processes have been discussed. By extension, we excluded papers related to architecture for business rule management, database management, and semantic web. We also did not include papers related to service composition, UML and software development. The second inclusion criterion includes any paper where a new modeling notation has been introduced, or any paper where an existing language notation has been used or extended.

As for exclusion criteria, we considered:

EC1: *Is the study a duplicate of an already considered study? (E)*

EC2: *Is the study less than 6 pages long? (E)*

EC3: *Is the study published after year 2000? (I)*

EC4: *Has the study passed the time-citation criteria? (I)*

EC5: *Is the full-text version of the study digitally accessible? (I)*

Duplicate papers were excluded from the list. If the duplicates are exact duplicates (same title and same authors), one of the duplicates is discarded. If on the other hand, the papers are version duplicates (original and revised), the older version is discarded and the most recent version is included in the list. The second exclusion criterion eliminates papers that are less than 6 pages long as they would hardly provide significant information. Any paper published before the year 2000 is removed from the list. The reason for this exclusion criterion is that we focused modeling notations used in modern BPM systems. We feel that papers published before 2000 are likely to provide no relevant information in the contemporary age especially with the technological advancements made since the turn of the century. The fourth exclusion criterion involves measuring the minimum number of citations that a paper published in a certain year needs to have in order to be considered in the literature review, as suggested by Pourmirza et al. [201]. The period of time under analysis was divided into two parts: papers published between 2000 and 2009, and papers that published after 2009. The average number of citations for each period as well as the standard deviation were calculated. In addition, the ratio (CV) of the standard deviation to the mean was also calculated. If the CV is less than 1, the period is considered to be fairly homogenous, thus, any paper within that period must have a number of citations higher than the average number of citations for that period. On the other hand, if the CV is higher than or equal to 1, then each year must be considered individually. So for a paper to be considered, the number of citations of that paper should be higher than the minimum between the average number of citations for the year in which the paper was published and the average number of citations for the 10 year period in which the paper was published. Furthermore, any paper published after 2014 was automatically considered. This is because not enough time has passed for those papers to gather a sufficient number of citations. Finally, for a paper to be considered, it must be accessible digitally. Accessibility here means that it is either available on the Internet or it can be accessed via a digital library with the use of a University subscription.

3.2.2.2 Screening Procedure

The papers were screened following the procedure provided in [6, 7]. The first step involves the application of inclusion and exclusion criteria defined earlier by examining the title and the abstract of papers. For a paper to be selected it must pass all the inclusion and exclusion criteria checking each criterion in descending order. This step was carried out by the author of this thesis.

Two reviewers, following the inclusion and exclusion criteria, screened the original list of papers again as recommended by Mistiaen et al. [8]. The inter-rate-agreement (Kappa value) was measured based on the three results to determine the extent to which they differ (Table 1). The original screening can be accepted if the percentage of papers selected by both the original screening and the two reviewers is greater than 0.6. The results of the inter-rate agreement were recorded to be

0.597 when comparing the original screening with the result from the first reviewer. Comparing the original with the second review yielded a Kappa value of 0.646. The mean value of the inter-rate agreement between the original screening and the two reviews was recorded to be 0.6215. For the sake of clarity, we also compared the results from the two subsequent reviewers and we got a Kappa value of 0.642. The reviewers discussed and came to consensus regarding disagreements. Note that we only applied the Kappa statistics to the inclusion criteria, as the exclusion criteria are straightforward and do not require any additional input or perspective from the researchers.

	First Reviewer	Second Reviewer
Original Screening	0.597	0.646
First Reviewer	-	0.642

Table 1. Result of Kappa measures for the three reviews

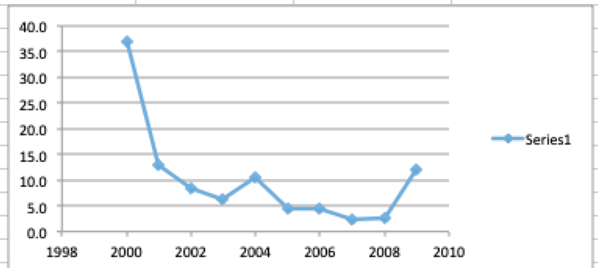
3.4 Search Results

The papers derived from each digital library and selected after applying the filtering criteria are:

- Scopus search for the combined strings resulted in 2,129 hits.
- Web of science yielded 813 hits.
- Duplicates were removed after combining the results from Scopus and Web of Science. The resulting count stood at 2,125.
- After removing papers with less than six pages, we got 1,733 papers.
- Papers published before 2000 were removed. This brought down the list to 1,701 papers.
- After applying the time-citation algorithm described above, the result was 920 papers.

The diagram below shows the result of applying the time-citation algorithm to paper published after the year 2000. We divided the time into 2 periods: 2000-2009 and 2010-2019.

Year	Number of papers	Total citations	Average citatios
2000	14	518	37.0
2001	28	362	12.9
2002	29	246	8.5
2003	46	294	6.4
2004	35	367	10.5
2005	69	315	4.6
2006	83	370	4.5
2007	71	168	2.4
2008	123	325	2.6
2009	111	1324	11.9
	609	4289	10.1
	Standard dev	10.16181249	
	CV	1.00365762	



Year	Number of papers	Total citations	Average citations
2010	117	941	8.0
2011	148	1589	10.7
2012	135	1103	8.2
2013	126	816	6.5
2014	127	524	4.1
2015	97	474	4.9
2016	129	372	2.9
2017	92	143	1.6
2018	99	56	0.6
2019	21	0	0.0
	1091	6018	4.7
	Standard dev	3.582874873	
	CV	0.755210169	

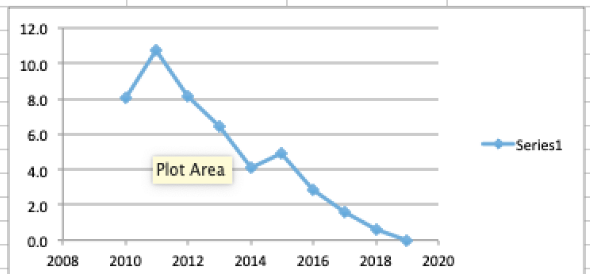


Figure 1. The result of the applying the time-citation algorithm to paper published after the year 2000

- The papers that were not available were removed from the list. This reduced the list to 791 papers.

- By applying the first inclusion criteria, we ended up with a list of 245 papers.

- Applying the second inclusion criteria brought down the list to 191 papers.

The table below shows how the papers were screened from the initial number to the final selection.

Step	Number of papers
Search	
Scopus	2,129
Web of Science	813
Total	2,942
Exclusion criteria (applied to 2,942) papers	
Removal of duplicates	2,125
Papers that are less than 6 pages	1,733

Papers published before the year 2000	1,701
Applying time-citation algorithm	920
Papers that are not available	791
Inclusion Criteria (applied to 791 papers)	
In the domain of BP and BR	245
Introducing, extending or applying a notation	191
Final papers used in the research	191

Table 2. Break down of the different stages of search and filtering

3.5 Data Extraction Strategy

The data extraction strategy ensures that the extraction of data is systematic and unbiased. We developed a data extraction table following the recommendations proposed by several studies [4, 7, 9].

Data Extraction Form

The data extraction form helps to extract information from the final list of papers. The form is composed of 3 sections. Each section consists of a set of data extracted from the paper. The first section consists of the paper's metadata such as the title, the author and the year of publication. The second section extracts data about the context of the study. This includes the industry, the objective of the study, findings and limitations. The final section extracts information useful to answer our research questions. This includes the notations being used, automation and tool support, etc.

Each paper was reviewed and the corresponding data columns in the table were extracted and filled accordingly. Another reviewer cross-checked the tables. All questions, discrepancies or differing views were resolved by consensus.

The data extraction table is shown in Table 3.

Data about the Paper	
Identifier	Unique id of the paper
Title	Title of the paper
Authors	Authors of the paper
Publication Year	Year of publication of the paper
Citations	Number of citations
Data about the Context of the Study	
Industry	Industry coverage of the paper
Study Objective	The objective of the application of business rules
Main findings	Main findings of the paper
Main limitations	Main limitations of the paper
Modeling notations data	
Notation	The notation being used.
Notation type	Introducing new notation, extending or applying an existing notation
Expressivity	The aspects of the process that is being captured.
Tool support	Tool support available for the notation
Automation/execution support	Whether or not the notation allows for automation
Modeling Scenario	Scenario where the rules have been applied
BR Integration	How the business rules are integrated within the model

Table 3. Data Extraction Table

3.6 Statistics

The final list of papers used in this research consists of 191 papers. The majority of the papers that have passed the criteria come from the last decade. This could be due to the fact that declarative modeling and business rule modeling has attracted the attention of researchers and business experts in the last decade. The figure below shows the distribution of papers across the years starting from the year 2000.

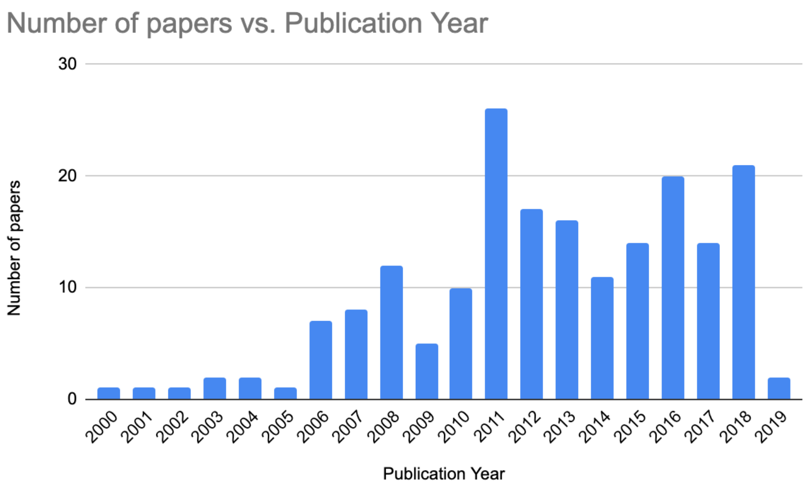


Figure 2. Number of papers extracted by year

4. Data Synthesis and Reporting

Analysis was carried on the data after data extraction. A framework for classifying how business rules are modeled was created. We used the data extracted from the papers in the structure shown in Table 3 to answer the research question. Through that, we were able to develop the classification framework.

4.1 Data Synthesis

We present here the results of the SLR by answering the research questions listed in Section 2. As stated in the main research question we try to explore the criteria that can be used as a basis for defining a framework for the modeling business rules. The rest of this section is divided into subsections with each subsection focusing on a different research question. We try to answer each question using the data extracted from the list of papers that we have selected in Section 3.

4.1.1 Notations (RQ1)

What notations exist that can be used to model business rules and decisions?

In recent years, there have been strides to come up with ways in which business rules can be incorporated into business process modeling. Prior to that, imperative approaches have mostly been used to model business processes. Nowadays, a number of notations exist that can be used to incorporate business rules of an organization in process models. There are other notations such eXtended Tabular Tree (XTT) [156] that are exclusively used to model business rules of an organization. These notations are sometimes used in combination with imperative notations, where business processes are modeled without explicitly defining the rules that govern these processes. Some of the available notations are discussed below.

Looking at the notations that use a declarative approach for business process modeling, Dynamic Condition Response graphs (DCR graphs) offer the user the ability to capture and manage business processes in a declarative fashion [15, 21, 19]. It is a constraint-based graphical notation that supports the capturing and continuous adaptation of business rules and business processes through immediate simulation. Like most declarative process modeling languages, DCR graphs capture the order of events and the rules that govern them [24]. Declare is another declarative process modeling approach that is used for modeling and executing loosely structured business processes [40]. Declare is the combination of a formal semantics based on Linear Temporal Logic (LTL) and a graphical representation for modeling [54]. Like DCR Graph, Declare focuses on modeling what should be done in a business process as opposed to how things should be done [40]. This is achieved by introducing rules in the model that constrain the execution of the process. The flow of activities in this case, is implicitly specified as any execution that is within the constraints without violating them [40, 26]. The Declare notation has given rise to other Declare-based notations. DeciClare [70], ARNE [78] and HiDec [79] are some of these notations. Others include Object-Centric Behavioral Constraint (OCBC) [82], and Multi-Perspective Declare (MP-Declare), an extension of Declare to model multiple perspectives such as data and time [27, 28, 37, 46 and 77].

Another declarative notation is Guard-Stage-Milestone (GSM) [165, 164, 166], a declarative modeling notation that is used to model case or artifact lifecycles [165]. Event-Condition-Action (ECA) [146, 145, 125] as well as other notations such as Drools [123, 124] and Event-Condition-Expectation [122], which follow the ECA approach to modeling business processes and business rules, are also declarative in nature. ECA, as the name implies, is a specification of business rules in a declarative fashion where the rule consists of an event that triggers the rule, a condition that is tested and if true causes the action to take place, and an action that is executed if the condition is true [123]. Alloy is also a declarative language for modeling software specifications and constraints [121].

Imperative or procedural modeling approach has also been used to model business processes with business rules. Here, the business rules are either implicitly modeled or modeled using a separate notation which the imperative modeling notation communicates with during execution. The most prominent imperative modeling notations are Business Process Modeling Notation BPMN and Petri Nets [150, 154]. BPMN is a graphical notation for the specification of business processes in the form of a process diagram. BPMN follows a similar pattern to the UML activity diagrams. Although BPMN notations are imperative, extensions of the notation such as BPMN2 [132, 138] and BPMN-D [137] have been known to support declarative specification of business rules while still partly maintaining the imperative approach. r-BPMN follows a hybrid approach to modeling where BPMN is integrated with REWERSE Rule Markup Language [139, 140], while in vBPMN, BPMN is combined with R2ML [141]. In [133, 134], business rules are represented in the form of a decision table schema, resulting in a BPMN model integrated with business rules.

Petri Nets is a modeling notation for the specification of distributed systems. It is a bi-directional graph consisting of nodes and places. The nodes are used to represent transitions and the places are used to represent conditions. Directed arrows represent which conditions (places) are conditional for the occurrence of which transitions (nodes). The most popular extension of Petri Nets is Colored Petri Nets (CPN). All the properties of Petri Nets are preserved in CPN. In addition, a distinction between different tokens is allowed in CPN. The color denotes a data value attached to the token. It is also possible to define complex types for colors, but tokens in a particular place are usually of the same type (color set). [Song et al, 154] proposed an extension to Petri Nets modeling notation by the addition of sensor data. The result is a notation called Context Aware Petri Nets (CAPN). The sensor data is represented as context information within the model. Other imperative notations include Conceptual Graph [167, 168] and BPEL [160, 64].

Another class of notations consists of those notations where business rules are modeled without the business process. Semantics of Business Vocabulary and Business Rules (SBVR) was introduced by the Object Management Group (OMG) as a standard for the specification of business knowledge and business rules. These rules are specified in controlled natural language that can be understood by humans and can be interpreted by the system [83]. [Bollen, 102] introduced SBVR and argued that most, if not all types of business rules can be modeled using SBVR.

The Web Ontology Language (OWL) [126 - 129] is a rule language for the specification of business knowledge or ontologies. A number of languages with varying syntaxes and specifications constitute the OWL, with formal semantics built upon W3C Resource Description Framework RDF. Another rule language Rule Markup Language or Rule Modeling Language (RuleML) is a family of rule languages for the specification of business rules using mostly an XML schema [130]. RuleML has been extended in the legal domain in [Governatori et al, 131] with the introduction of LegalRuleML, which can be used to specify Legal knowledge and support for legal requirements. eXtended Tabular Trees (XTT) [156] is a specification based on Attribute Logic for the representation of business knowledge and business rules. The notation provides a visual representation of rules in a logical and efficient way. XTT differs from other languages in that it has an explicit structure as opposed to flat rule sets. The structure consists of tables (representing groups of rules) and inter-table links (allowing switching control between tables) forming a decision tree structure [157, 159]. Other BR languages include Fuzzy Control logic (FCL) [169, 170] and Semantic Business Process Modeling Language (SBPML) [171, 172].

So far, we have identified different types of notations. We have also identified the different modeling approaches that the notations follow. We have identified declarative notations such as Declare and DCR graphs. We were also able to identify imperative notations such as Petri Nets and BPMN. Another approach that was identified is the conditional or business rules modeling approach where the notations are used to represent business rules exclusively without specifying other aspects of the business process. This approach is seen in notations such as XTT and RuleML. The table below shows the notations and the papers where the notations have been studied. This lays the groundwork for identifying our framework and provides a basis for answering the other research questions.

Notation	References
DCR Graphs	15,16,17, 19, 20, 21, 22, 23, 24, 25
Declare	26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 39, 40, 41, 42, 43, 44, 45, 47, 48, 49, 50, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 78, 79, 80, 81, 82, 142, 143, 144, 145, 146, 148
MP-Declare	27, 37, 38, 46, 51, 77
SBVR	83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 95, 96, 97, 98, 99, 100,101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118
Alloy	119, 120, 121
ECA / Drools	146, 145, 183, 122, 123, 124, 125
OWL	126, 127, 128
RuleML	129, 130, 131
BPMN	132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 94
Petri Nets	150, 151, 152, 153, 154, 155
XTT	156, 157, 158, 159, 160
BRL	161, 163, 163
GSM	164, 165, 166
FCL	169, 170
SBPM	171, 172
Conceptual Graph	167, 168

Table 4. Languages / Notations and their reference

4.1.2 Expressivity (RQ2)

What perspective can the business rule modeling notations capture?

This section explores the expressivity of the notations. By answering this question, we provide an overview to the user on the capabilities of the notations being examined. This will help the user in understanding what aspects of the business process can be modeled using each notation. Some notations allow the representation of data, resources and actors while some of these features are not captured in other notations. The intended use case of the user and the aspects of the business process that need to be captured will determine which notation is best suited for their use case. Below we give a description of the aspects of the business process captured by the identified notations and how these aspects are represented in using these notations.

DCR Graphs has no data representation [15]. The graphical notation allows the user to specify activities or events (a part of an activity that the system can keep track of [19]), represented as boxes [15, 20, 21]. The activities are annotated with names and actors (role or participant) or principals (persons, processors, services) [16]. Roles are also assigned to principals [16]. Each activity has a color that determines its status. Grey means that the activity is not currently executable, red means the activity is required and greyed out means the activity is not executable anymore [15]. An Activity represented with dashed boxes is excluded initially and could be included by another activity [22]. This means that the inclusion of that activity is dependent on the execution of another activity. that Activities can be assigned to multiple roles [17]. DCR Graph activities are constrained by activity relations or constraints [17].

Another declarative notation that has taken the attention of researchers in recent years is Declare. Activities and constraints are the components of a Declare model. Constraints can be mandatory, represented by continuous line, and thus must be fulfilled, or optional constraints, which can be violated [41]. Activities represent the tasks within the process at an atomic level. Constraints constrain the activities and determine the way in which these activities are to be executed [29]. Declare consists of over 20 different constraint templates from which business rules can be modeled in the form of dependencies between activities in the business process [26]. These templates are grouped into three sets, existence, negation and relation templates. Existence or cardinality constraints are modeled as numbers above the box that represents the activity with cardinality (e.g., 0; N..M). These numbers specify how many units of that activity must be executed [29]. Relation and negation constraints are represented graphically using a set of annotated links between activities. The whole semantics of the notation is captured in Linear Temporal Logic (LTL) [26]. A number of notations have been derived from Declare, giving the notation a more robust expressivity and improving the way the modeler uses the notation.

MP-Declare is an extension of Declare that can be used to model constraints over both control-flow data flow as well as time dimensions of a process [37, 77]. The formal semantics of MP-Declare is based on Metric First-Order Linear Temporal Logic (MFOTL). DeciClare [70] is an extension of Declare that can be used to capture the time perspective, the resource perspective and the data perspective of a business process. Time Declare is an extension of Declare that captures the time perspective where all constraints are annotated with interval times specifying the time distance between activity occurrences [76]. For example, timed response between two activities A and B specifies that when activity A occurs activity B should eventually follow within the specified time interval.

In SBVR, business knowledge is represented in two parts: 1. Representing business vocabularies, 2. Representing business rules [90]. Business vocabulary consists of a set of concepts that can be used to describe how to conduct a business. These concepts are specified using names, verbs etc. [90]. Noun concepts or Object types [85, 89, 95] are used to represent objects in the business process such as Order, Employee, etc. [84, 85, 95]. An individual concept represents a qualified noun or an instance of the Object type, e.g., "Birmingham" [85, 89, 95]. The relationships between concepts are represented as Fact types, e.g, Order contains OrderLine [84]. Fact types are constrained by business rules that build upon these fact types: Order contains at least one OrderLine [85, 89]. These rules can be behavioral (relating to how to execute a business entity) or structural (dealing with organizational setups) [89]. A Characteristic, associated with an object, represents the abstraction of a property of that object. For instance, the name of a bank, where name is a characteristic of the object type bank [85, 89, 107]. SBVR concepts are represented using different fonts. Noun concepts are represented using the term font (green). Blue is the verb font used to represent verb concepts. Linguistic particles such as prepositions and conjunctions, which help to give meaning to statements, are represented using the keyword font (red) [84]. In addition to these concepts, SBVR also supports the representation of deontic and alethic logic [86, 98]. Restricted qualifications such as at most and at least, are also supported in SBVR. In addition to specifying business rules, an SBVR vocabulary can also be used to represent roles, data objects or activities without any business rules specified over them [106]. There is no specific syntax that is enforced in SBVR but it is flexible and allows different notations [106]. Since meanings are separated from representations in SBVR, it is possible to have several representations for one meaning. These representations can be words, sounds or figures [84].

SBVR was extended in [84, 115] so as to support representation of agents or actors, events, activities and process statements. The result was a framework called EM-BrA^{2}CE, where these concepts as well as all the other SBVR concepts can be modeled. In EM-BrA^{2}CE, a business process instance consists of activities. These activities have a specified duration and are performed by an agent. Events reflect the change in the lifecycle of an activity [83, 115]. EM-BrA^{2}CE also supports the representation of control-flow and organizational aspects as well as data constraints [84].

Alloy can be used to model data (sets and relations), formulas or rules, which are used to constrain the data, and provides a set of commands that invoke the Alloy analyzer to check the validity of those rules [119]. Formulas are structured with the help of predicates, which are parameterized boolean functions. Alloy has a syntax similar to OCL, but the main difference between Alloy and OCL is the presence of imperative elements in OCL, whereas Alloy is fully declarative [121].

ECA rules capture the dynamic behavior as well as rules in a business process [146]. The extension of this notation ECAA, proposed in [146], captures alternative actions in case the condition is not fulfilled. Linking business rules allows modeling sequential actions in ECAA [146]. An event is raised once the preceding action is complete triggering the current action, thus allowing linking business rules. Parallel actions (AND split) alternative actions (OR split) and iterations of actions (loop) are also captured by ECAA. Other modeling options captured by ECAA are event sequences, periodical events, event selection and time. The time can be absolute, relative or repeated. The details of these components can be found in [146]. ECAA also allows modeling actors (persons and application systems) and data. ECAPE (event-condition-action-post-condition-event), an extension of ECA allows the modeling of post-conditions that are evaluated after the action in an ECA rule has been completed [147]. ECA can also be used to capture time constraints [184] between events, a feature that is very important especially in financial trading [184]. To deal with expectations and implement a new approach (Event-Condition-Expectation) based on Drools' Event-Condition-Action, the work presented in [122] exploited drools fusion. Drools Fusion is capable of modeling different aspects of a business process. Fact classes can be annotated with meta-data, which the engine considers as payloads of events [122]. The engine also manages temporal information, which can be passed as part of class instance fields or by the engine. In addition to managing events and conditions and performing the required actions, Drools Fusion was extended in

[122] to handle and manage expectations. Expectations are defined in [122] as the expected behaviors resulting from triggering of an event. The resulting actions are checked for fulfilment or violation of the expectations. The drools rule engine takes an object-oriented approach inspired by the RETE [123] algorithm. Custom operators and fuzzy logic are also supported by Drools [123].

In [126], OWL-S was used for its rich semantics that can be used to describe web services. It consists of a profile, a process model and a grounding. The profile gives a description of what the service does [126]. The process model describes how it works and encompasses all the components of a process such as input and output, conditions, and results. The grounding on the other hand specifies the manner in which the web service can be accessed. [126] argued that the OWL notation in comparison to RDFS (Resource Description Framework Schema), offers a much higher expressive power for the description of complex processes. In addition, all the business rules relating to the healthcare domain can be modeled using OWL, according to [126]. In particular, all the resources related to the healthcare domain can be captured using this notation including people and assets. OWL also supports the definition of properties for data and objects. Table 5 provides a description of these properties. OWL 2, an extension of OWL, supports data types and has an extended collection of data types [202].

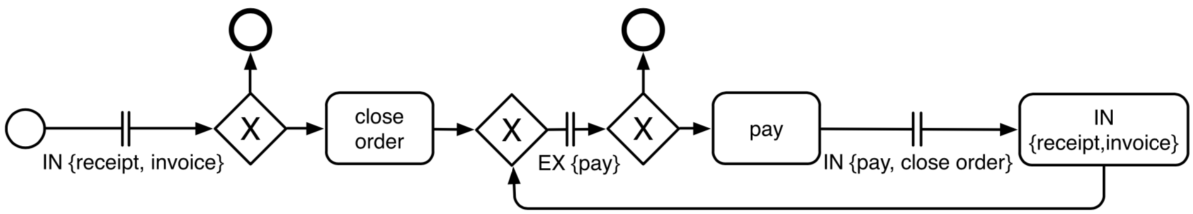
Data Property	Domain	Range
hasBusinesObject	Process	BusinessObject
hasException	Process	Exception
hasPerformanceMetric	Process	Metric
hasStartEvent	Process	StartEvent
hasEndEvent	Process	EndEvent
hasQueueEvent	Process	QueueEvent

Object Property	Domain	Range
hasTargetValue	Metric	Float
hasUpperControlLimit	Metric	Float
hasLowerControlLimit	Metric	Float
hasMetricValue	Metric	Float
hasBusinesObjectID	BusinessObject	String
hasTimestamp	Event	DateTime

Table 5. Owl’s Data properties and Object properties [202]

RuleML’s XML syntax can be used for knowledge representation [129]. RuleML is capable of capturing derivation rules (derived from other statements or calculations), integrity rules (assertions that must be fulfilled during execution) or reaction rules (conditions to be executed when a certain event occurs) [129]. RuleML also captures objects, entities and resources [130]. RuleML was extended in [131]. Legal metadata and legal operators (deontic and behavioral) operators were added to RuleML. The extension allows the representation of all Legal knowledge and support for legal requirements.

Although not explicitly covering business rules, BPMN is the most extensively studied notation with a lot of resources available for the notation. BPMN2 [132] can be used to model temporal dimensions that specifies the time needed to execute an activity. Other aspects such as events, control flow and information flow can also be modeled. In addition to duration, the start and end time of an activity can also be modeled. Time duration cycle and dates are supported as timer events in BPMN2 [94]. BPMN captures events, flow connectors, activities and choices of a business process. An activity represents an execution step and is modeled as a rounded rectangle in the graphical notation [137]. Flow connectors, represented as edges, represent the ordering of nodes. The same elements are inherited by BPMN-D [137]. In addition, connectors in BPMN-D could also mean that other activities can be performed between the two activities being connected by the arc ("loose" ordering). Two of the loose connectors are inclusive flow and exclusive flow, which are differentiated by the annotation "in" and "ex" respectively and a set of tasks T that are to be executed 0 or more times while traversing the arc from one node to another [137] (see figure 7 below).



Notation	Name	Semantics
t	Atomic task	As in BPMN: perform t
IN $\{t_1,\ldots,t_n\}$	Inclusive task	Perform a task among $t_1, \ldots, t_n$
EX $\{t_1,\ldots,t_n\}$	Exclusive task	Perform a task different from $t_1, \ldots, t_n$
ANY	Any task	Perform any task from those available in the business context

Figure 3. BPMN-D activity nodes

rBPMN combines BPMN using its rich syntax in representing business processes and its wide acceptability and R2ML [140]. BPMN is used to model events, activities and gateways. Gateways are used to represent decisions for merging and splitting control flow. In addition pools representing participants are also modeled using the BPMN notation [140].

Colored Petri Nets [150] combines the modeling capabilities of Petri Nets and the CPN ML programming language. Petri nets are graphs of nodes. The nodes consist of places and transitions. A place represents a condition and a transition represents an event [151]. An arc in the form of a directed edge connects a place and a transition. Token distributions over places represent markings. When all the input places of a transition contain a token, it becomes enabled and can be fired. A token is moved from each input place into the output places of the transition when the transition is fired [151].

XTT2 is a notation for the representation of business knowledge and inference control. The notation provides visual and logical representation of business rules [156]. Rules in XTT2 are represented in groups with shared context into decision tables that are interconnected with each other [156, 157, 158, 159, 160].

The GSM BA (Business Artifact) is an approach for modeling decision intensive business processes using the GSM framework, where both data and process are integrated into one unified framework [164]. This approach supports parallelism and the representation of activity lifecycles. GSM is a declarative approach to process lifecycle modeling. The lifecycle in GSM consists of stages. Each stage in the lifecycle consists of activities or tasks. A stage can be atomic, where tasks are stored without nesting; or composite where sub-stages are nested within the composite stage [164]. Some of the tasks supported within a stage include assignment, tasks performed by humans, creation of BA instances, and service invocation. In each stage, there is at least one milestone and one guard. A GSM model consists of two attributes: data and status. Status represents the status of a milestone or a stage [165, 166]. A stage is either open or close depending on the Boolean value of its status. The status of a milestone can either be true or false representing whether the milestone has been achieved or not. Business rules are captured using the ECA rule definition, which defines for each event, which conditions causes the status to change, invoking an action [165, 167]. The EC part of a rule in GSM is called sentry [166, 167], which can either be plus sentry indicating that a milestone has been achieved or a stage has become open [165, 166]. Events can be distinguished as either internal or external [165].

Conceptual graphs (CGs) offer the capabilities of modeling business processes at different levels [167]. They can serve as a language at conceptual level that system analysts and business representatives can use to communicate and share ideas using their rich, simplified graphical schemas. At implementation level, CGs can be used for representing different components of a business workflow system ranging from databases, knowledge and inference engines [167]. CGs can also be mapped to other first logic standards and can also serve as a connecting tool between different types of visual, formal and natural languages [167]. A CG consists of concepts and relations. An arc links a concept with a relation, but cannot be used to link two concepts or two relations directly. The simplest form of conceptual graph given in [167] is “A cat is on a mat" represented as [Cat]− > (On)− > [Mat]. Cat and Mat are concepts and are linked by the relation “On”. The concepts have types and a referent. For instance, cat is the concept type and the referent for that concept is blank. The referent represents the quantifier and a blank referent denotes existential quantifier. Relations in CG also have types along with a valence, which must be a positive integer. In the example above, the type is “On” and the valence has a signature <Cat, Mat>, with a value of 2 [167]. Graphically, concepts are represented as rectangles and relations are shown as ovals [167]. Actors are represented as diamonds.

Fuzzy Control Logic (FCL) was used in [169] to capture the norms of a business process or the conditions upon which the process needs to be executed. FCL is used to capture deontic logic, which includes permissions, obligations, negations and violations [169]. The structure of an FCL rule is r: A1,…,An =>B. r represents the name of the rule, A1,…,An represent the premises, B represents the conclusion [169]. FCL also handles a superiority relation between rules by determining the relative strengths of the rules and prioritizing the stronger rule over the weaker one in cases of conflicts in the conclusions [169].

Notation	Aspects of BP captured by the notation
DCR Graphs	- Activities - Actors, roles - Activity relations / constraints - Events - Activity nesting
Declare MP-Declare	- Activities, - Constraints based on constraint templates - Temporal perspective - Optional Behaviors - Resources (DeciClare) - Hierarchical dimension - Data perspective (MP-Declare, DeciClare)
SBVR	- Business vocabulary (Objects, fact types) - Business rules - Actors, events, activities and process types (EMBRACE) - Time (EMBRACE) - Control flow and data (EMBRACE) - Deontic and alethic logic

	- Data types and activity type
Alloy	- Data - Rules - Time - Objects
ECA	- Events - Activity - Rules - Event sequence - Time - Actors - Data
OWL	- Rules - Resources (actors, assets) - Data - Object types, data types
RuleML	- Deontic and behavioral rules
BPMN	- Events, activities, choices, flow connectors - Temporal aspect (BPMN2) - Business rules (rBPMN) - Actors and resources
Petri Nets	- Business rules (CPN) - Events and conditions
XTT	- Business rules
GSM	- Data - Activities - Constraints (ECA) - Events
FCL	- Deontic logic
SBPM	-
Conceptual Graph	-Concepts and relations

Table 6. Notations and the components they capture

In this section, we have looked at the notations we derived from the literature in the previous section and examined what aspect of the business process and business rules they can capture. We also provided an overview in the form of a table describing each notation and what aspects it captures. This will give the user a summary of what the notations can capture and which notation is best suited for their individual needs.

4.1.3 Business rules representation (RQ3)

How are these rules integrated within business process models?

We have so far identified notations that are used to capture business rules. We have also looked at the expressive capabilities of these notations. This question is now focusing on how business rules are integrated into business process modeling notations. In some languages, the business rules are expressed in terms of business process models. Other notations represent business rules in a way that they are supposed to be connected to a process model. Finally, some notations do not have a mechanism for representing business rules at all, but rely on other notations to represent these rules. Answering this question will help the user in deciding how to handle their business rules, either as a part of a business process model or as a separate entity.

DCR graphs, like any declarative process modeling notation, can be used to model business processes along with business rules. The rules or constraints determine what is not to be violated in the process execution [15, 16, 17]. The way how to execute the process is left to the user as long as the constraints are not violated. Business rules or constraints are represented in DCR graphs as arrows linking activities together. Condition, response, exclusion and inclusion constraints can be modeled [15, 23, 24]. A Conditional constraint is represented as a directed arrow with a dot at the tip (A□□B). A is a condition for B if for activity B to occur A must have occurred before. Response constraint is represented as a directed arrow with a dot preceding the arrow (A□□B). In this case, activity B must occur with every occurrence of activity A. Exclusion constraint is represented as (A□%B), which means activity A excludes activity B. Inclusion on the other hand is represented as (A□+B), which means when activity B cannot be executed until activity A occurs which includes it. Finally, A□wB denotes a milestone response, meaning that A has B as a response but, in this case, B cannot be executed if A is not [16].

In Declare, business rules are represented as constraints that are generated from constraint templates. These templates form the grouping upon which business rules are classified and integrated into a declarative business process model. In [26], the templates are grouped into three main classes: existence, relation and negation templates. Business rules are modeled as a relationship between tasks. Each constraint has a value on a process execution that can be either true if the constraint is fulfilled, or false in case of violation of the constraint [26]. The constraints are usually modeled graphically as edges linking one activity to another. An example is the response constraint, which specifies that activity B must be executed eventually after every occurrence of activity A [26]. Figure 4 provides the semantics of all Declare constraints templates and their graphical representation.

TEMPLATE	NOTATION	LTL FORMALIZATION	INEQUALITY CONSTRAINT
init(A)	init A	A	$(\#A_T \geq 1)$
existence(A)	1..* A	$\Diamond A$	$(\#A_T \geq 1)$
absence(A)	0 A	$\neg\Diamond A$	
choice(A, B)	A B	$\Diamond A \vee \Diamond B$	$(\#A_T \geq 1 \wedge \#B_T \geq 1)$
exclusive_choice(A, B)	A B	$(\Diamond A \vee \Diamond B) \wedge \neg(\Diamond A \wedge \Diamond B)$	$(\#A_T \geq 1 \wedge \#B_T \geq 1)$
responded_existence(A, B)	A B	$\Diamond A \rightarrow \Diamond B$	$(\#A_T = 0 \vee \#B_T \geq 1)$
co_existence(A, B)	A B	$(\Diamond A \rightarrow \Diamond B) \wedge (\Diamond B \rightarrow \Diamond A)$	$(\#A_T \geq 1 \wedge \#B_T \geq 1)$
response(A, B)	A B	$\Box(A \rightarrow \Diamond B)$	$(\#B_E \geq \#A_T)$
precedence(A, B)	A B	$\neg B \mathcal{W} A$	$(\#A_E \geq \#B_T)$
succession(A, B)	A B	$\Box(A \rightarrow \Diamond B) \wedge (\neg B \mathcal{W} A)$	$(\#A_E = \#B_T)$
alternate_response(A, B)	A B	$\Box(A \rightarrow \bigcirc(\neg A \mathcal{U} B))$	$(\#B_E \geq \#A_T)$
alternate_precedence(A, B)	A B	$(\neg B \mathcal{W} A) \wedge \Box(B \rightarrow (\neg B \mathcal{W} A)))$	$(\#A_E \geq \#B_T)$
alternate_succession(A, B)	A B	$\Box(A \rightarrow \bigcirc(\neg A \mathcal{U} B)) \wedge (\neg B \mathcal{W} A) \wedge \Box(B \rightarrow (\neg B \mathcal{W} A)))$	$(\#B_E = \#A_T)$
chain_response(A, B)	A B	$\Box(A \rightarrow \bigcirc B)$	$(\#B_E \geq \#A_T)$
chain_precedence(A, B)	A B	$\Box(\bigcirc B \rightarrow A)$	$(\#A_E \geq \#B_T)$
chain_succession(A, B)	A B	$\Box(A \rightarrow \bigcirc B) \wedge \Box(\bigcirc A \rightarrow B)$	$(\#B_E = \#A_T)$
not_co_existence(A, B)	A B	$(\Diamond A \rightarrow \neg\Diamond B) \wedge (\Diamond B \rightarrow \neg\Diamond A)$	
not_succession(A, B)	A B	$\Box(A \rightarrow \neg\Diamond B)$	
not_chain_succession(A, B)	A B	$\Box(A \rightarrow \neg \bigcirc B)$	

Figure 4. Declare constraint templates [26]

In Figure 4, Init(A) and End(A) specify respectively that task A must be the first and the last activities to be executed in a process instance. Absence(A) means that A must not occur in the process instance. Choice (A, B) ensures that either or both A and B occur eventually in the process execution. The complete description of all the constraints can be found in [26].

In Timed-Declare [45, 54, 60], the constraints are annotated with time intervals. In Figure 5, the response constraint $G(A \rightarrow F_{[t1,t2]}B)$ implies that the occurrence of A at a time t necessitates the occurrence of B within the interval [t+t1, t+t2]. In [45], it is noted that negative constraints can be used to express latencies, whereby the accomplishment of an activity requires at least a minimum amount of time to be executed. A case like this is the not response $G(A \rightarrow \neg(F_{[0,t1]}B))$. Here, if activity A is executed at a specific time t, then activity B must not occur in the interval [t, t + t1] [45]. A more detailed representation of all the Timed-Declare constraints is found in [54].

constraint	untimed semantics	timed semantics	notation
responded existence	$\mathbf{F}A \rightarrow \mathbf{F}B$	$\mathbf{G}(A \rightarrow (\mathbf{O}_{[t_1,t_2]}B \vee \mathbf{F}_{[t_1,t_2]}B))$	A $[t_1,t_2]$ B
response	$\mathbf{G}(A \rightarrow \mathbf{F}B)$	$\mathbf{G}(A \rightarrow \mathbf{F}_{[t_1,t_2]}B)$	A $[t_1,t_2]$ B
precedence	$\mathbf{G}(B \rightarrow \mathbf{O}A)$	$\mathbf{G}(B \rightarrow \mathbf{O}_{[t_1,t_2]}A)$	A $[t_1,t_2]$ B
alternate response	$\mathbf{G}(A \rightarrow \mathbf{X}(\neg A\mathbf{U}B))$	$\mathbf{G}(A \rightarrow \mathbf{X}(\neg A\mathbf{U}_{[t_1,t_2]}B))$	A $[t_1,t_2]$ B
alternate precedence	$\mathbf{G}(B \rightarrow \mathbf{Y}(\neg B\mathbf{S}A))$	$\mathbf{G}(B \rightarrow \mathbf{Y}(\neg B\mathbf{S}_{[t_1,t_2]}A))$	A $[t_1,t_2]$ B
chain response	$\mathbf{G}(A \rightarrow \mathbf{X}B)$	$\mathbf{G}(A \rightarrow \mathbf{X}_{[t_1,t_2]}B)$	A $[t_1,t_2]$ B
chain precedence	$\mathbf{G}(B \rightarrow \mathbf{Y}A)$	$\mathbf{G}(B \rightarrow \mathbf{Y}_{[t_1,t_2]}A)$	A $[t_1,t_2]$ B

constraint	untimed semantics	timed semantics	notation
not responded existence	$\mathbf{F}A \rightarrow \neg\mathbf{F}B$	$\mathbf{G}(A \rightarrow (\neg\mathbf{O}_{[t_1,t_2]}B \wedge \mathbf{F}_{[t_1,t_2]}B))$	A $[t_1,t_2]$ B
not response	$\mathbf{G}(A \rightarrow \neg(\mathbf{F}B))$	$\mathbf{G}(A \rightarrow \neg(\mathbf{F}_{[t_1,t_2]}B))$	A $[t_1,t_2]$ B
not precedence	$\mathbf{G}(B \rightarrow \neg(\mathbf{O}A))$	$\mathbf{G}(B \rightarrow \neg(\mathbf{O}_{[t_1,t_2]}A))$	A $[t_1,t_2]$ B
not chain response	$\mathbf{G}(A \rightarrow \neg(\mathbf{X}B))$	$\mathbf{G}(A \rightarrow \neg(\mathbf{X}_{[t_1,t_2]}B))$	A $[t_1,t_2]$ B
not chain precedence	$\mathbf{G}(B \rightarrow \neg(\mathbf{Y}A))$	$\mathbf{G}(B \rightarrow \neg(\mathbf{Y}_{[t_1,t_2]}A))$	A $[t_1,t_2]$ B

Figure 5. Timed-Declare constraints templates [45]

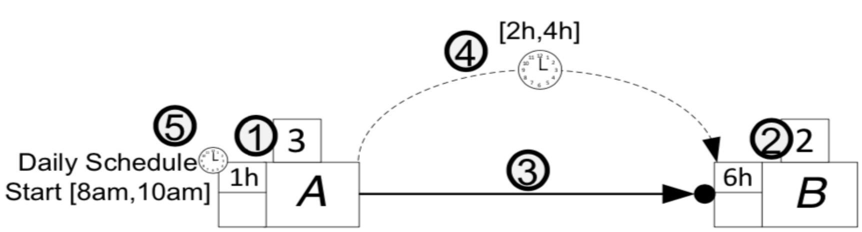


Figure 6. A simple TConDec-R Process Model [32]

In [32], the notion of temporal constraints was introduced. Figure 6 gives an overview of some constraints expressed in TConDec-R, the notation used in [32]. This diagram (represented as Acts={(A,1h),(B,6h)}) is a simple TConDec-R model consisting of two activities, A and B. Each activity is annotated with an estimated duration. Activity A takes 1 hour and activity B takes 6 hours. The constraints in this model are: (1) Exactly(A, 3) and (2) Exactly(B,2), which are cardinality constraints stating that A must be executed three times and B must be executed twice, (3) Precedence(A,B), stating that A must be executed before B. Another constraint in the model is (4) TimeLagEndStart(A,B,2h,4h), which means that for each execution of an instance of A, there must be an instance of B executed with a time lag between 2 and 4 hours. The constraint DailyScheduleStart(A,8am,10am) signifies that each execution of A must be started between 8 and 10 am.

In SBVR according to [84], *rules are always constructed by applying necessity or obligation to fact types*. Business rules are integrated into a model as constraints over fact types [84]. Each business rule is formulated based on at least one fact type and can be organizational or behavioral [85, 89, 100, 101, 107]. According to [96], applying obligation or necessity to a fact type generates a business rule. For instance the statement "Order must contain at least one OrderLine" is a business rule [84]. To model business rules in SBVR, there are a number of keywords that are built-in within the notation that combine term, verb, name and fact type to form a business rule [90]. These words are categorized, according to [90], into qualification, modal and logical keywords. Figure 7 shows a simple business rule stated in SBVR-SE (SBVR- Structured English) [92].

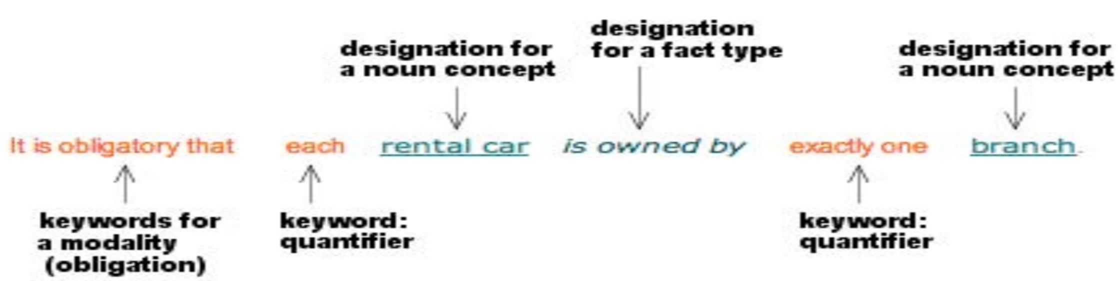


Figure 7. Different components of an SBVR statement [92]

In [93], a tabular description of the different types of business rules in SBVR was presented. A classification of business rules into structural or organizational (alethic) and behavioral (deontic) was presented as shown in Table 8. Along with that, examples were also given.

	Modality Type	Example in SBVR Structured English
Structural (alethic)	necessity	**An** order **always** *has* **exactly one** customer.
	non-necessity	**It is not necessary that an** order *has* **a** customer**.**
	possibility	**It is possible that an** order *has* **more than one** line item.
	impossibility	**An** order **never** *has* **more than one** customer.
	contingency	**It is possible but not necessary that an** order *has* **a** customer**.**
Behavioral (deontic)	obligation	**Each** order **must** *be processed* **within** one business day.
	non-obligation	**It is not obligatory that a** customer representative *approve* **a** credit.
	permission	**A** customer representative **may** *approve* **a** credit.
	prohibition	**A** clerk **must not** *change* **the** terms and conditions.
	optionality	**It is permitted but not obligatory that a** customer representative *approve* **a** credit.

Table 7. Deontic and Alethic rules in SBVR [93]

Business rules are represented as assertions in Alloy. These formulas serve to constrain the data within the model [119]. The rules are presented as assertions and they use functions and facts within the model to constrain the model [120]. An example of a simplified alloy formula is given in Figure 8 below. The model is used to fetch a page from its owner or from cache. The detailed explanation of the function is found in [120]. The main point of interest here is the assertion which represents business rule. The expectation is that every time the function is called, a fresh page is returned. The rule in this regard ensures that the new page is checked for freshness.

```
module webcache

sig Time {}
sig URL {}
sig Server {page: Time -> URL ->? Page} part sig Cache, Owner extends Server
{} sig Page {life: set Time}

fact OwnerFreshness { all s: Owner, t: Time,
      u: URL, p: Page |
      (t -> u -> p) in s.page =>
  t in p.life }

fun Get (t,t':Time, c:Cache, o:Owner, u:URL, p:Page) {
  c.page[t'] in c.page[t]
  + o.page[t] -
    {u:URL, p:Page |
     t' not in p.life} p in
```

```
      (c+o).page[t'][u]   }
assert Freshness {
all t,t':Time, c:Cache,
o:Owner, u:URL, p:Page | Get (t, t', c, o, u, p) => t' in p.life }

check Freshness
```

Figure 8. A simple Alloy formula [120]

Drools representation of business rules takes the ECA rule approach [123, 124], where each event is evaluated against a given condition. If the condition holds, then an action is taken otherwise an alternative action is taken. This rule modeling approach was extended in [123] to support expectations where an expected outcome is already decided so the action can be checked for fulfilment or violation of the expectations. An example of ECA rule in drool is given in figure 9 below that catches rule violation.

```
declare Message @role(event) end
rule "CEP Rule Example"
when
     $m: Message( $s: sender, $r: receiver, content == "HELO")
     not Message ( sender == $r , receiver == $s ,
                   content == "+ACK", this after[0,5s] $m )
then
     log("Acknowledgement expected, not received in 5
    seconds");
end
```

Figure 9. A simple Drools ECA rule [122]

In the OWL notation, business rules are represented as relations between Actions [127]. The rules are represented in OWL as properties and can be one of three constraint types [127]: hasDependency, which is an object property that stipulates that if another action has been executed, the action, which possesses the property can also be executed. hasExclusionInParallel prevents two actions from being executed in parallel. hasExclusionInSequency prevents two actions from being executed in sequence [127]. Derivation business rules [127] or low-level business rules [126] representing business logic are represented in the OWL ontology. These types of rules have been represented in [126, 127, 201] using the Semantic Web Rule Language (SWRL) [127]. In [128], rules were represented using SWRL as well as OWL. Below is the representation of a rule in OWL language, where the composition of the parent property and the brother property results in the uncle property. That is the brother of a parent is an uncle. The variables are conventionally prefixed with a question mark. The left hand side of the equation is called the antecedent and the right hand side is called the consequent. This forms the “IF something DO something” logic representations [127].

parent(?x,?y) ∧ brother(?y,?z) ∧=> uncle(?x,?z)

In RuleML, business rules are represented as XML constraints stored in a rule base of the rule engine [129]. The rule representation is RuleML follows the IF-THEN format where an action is taken if a condition is fulfilled or an assertion becomes true if a condition is fulfilled. The rule “*A customer is “Premium” if their spending has been min 5000 dollars in the previous year”* is represented in RuleML format as follows:

```
<Assert mapClosure="universal">
        <Implies timesBlock="#t2" ruleType="defeasible" id="rule1">
                <then timesBlock="#t1">
                        <Atom id="atm1">
                                <Rel>premium</Rel>
                                <Var>customer</Var>
                        </Atom>
                </then>
                <if timesBlock="#t1">
                        <Atom id="atm2" timesBlock="#t3">
                                <Rel>previous year spending</Rel>
                                <Var>customer</Var>
                                <Var>x</Var>
                                <Data>= 5000$ </Data>
                        </Atom>
                </if >
        </Implies>
</Assert>
```

In the BPMN notation, OCL notation has been used in [132] to model business rules. OCL expressions represent business rules and are used as constraints over BPMN classes [132]. [133] used the decision table schema to model business rules which are applied over the model during execution. These rules are represented using the XTT2 modeling notation. An XTT2 rule is defined in [133] as the equations:

$$r_i = (cond_i, dec_i)$$

$Cond_i$ represents the conditional part of the rule consisting of atomic formulas and dec_i represents the decision part also consisting of valid atomic formulas. The decision component of XTT2 also called the XTT2 decision table table is defined as $t = (r_1, r_2, \ldots, r_i)$

A decision tree [134] was constructed and used to capture business rules applied to a BPMN model [134]. SBVR has also been used to capture business rules over BPMN processes in [135, 94]. REWERSE Rule Markup Language (R2ML), a rule language was used in [139, 140] to capture business rules in BPMN processes. The XML graphical aspect of R2ML makes it possible to use it along with the BPMN notation [139].

In [150], business rules are modeled as XML inside the RULE file of the Colored Petri Nets model. SBVR vocabulary was used in [155] to model business rules in the Petri Nets models. In [151], business rules were represented using the If…then rule format. From these rules, a Petri Nets model is generated. In XTT2, business rules are represented using a decision table structure [156,157]. Similar rules are grouped together in the same table [157]. These tables are interconnected together within the system [156, 157, 158].

The ECA rules definition is used to model business rules in GSM [165, 165, 166]. The ECA rules form the logic or the guard of the stages. The guard determines whether a stage is open or closed [164]. Once a guard or rule gets triggered, the stage becomes active or open and the operations within an atomic stage become initiated. The sub-stages within a composite stage also become ready to be activated [164]. The EC (Event-Condition) part of the rules is called sentry [165, 166]. A rule is composed of a sentry and an internal action [166]. Following the ECA rule format, when an event is triggered, the condition is checked. If the condition holds, then the internal action is executed. This action can be the opening or closing of a stage, or reaching a milestone [166]

Originally, Conceptual Graphs (CGs) were not used to model business rules but were used as part of a framework where CG serves as the notation for the representation of the business and information system components [167]. Business rules were stored in an active database in XML format [167]. In [168], CG was extended with the addition of a rule base in the knowledge base component for the storage and management of business rules [168].

Fuzzy Control Logic (FCL) was used in [169] to model business rules in the form of deontic logic. FCL is used to capture deontic logic, which includes permissions, obligations, negations and violations [169]. FCL rules are represented as norms in the form of r: $A_1, \ldots, A_n \Rightarrow B$, where r is the name of the rule which is unique, $A_1, \ldots, A_n$ form the premise of the rule and B represents the conclusion. The premises are built from the following logical operators: ¬(negation), O (obligation), P (permission), and ⃠(violation/reparation) [169].

This section gave an overview of the different ways in which business rules can be represented and integrated into business process models. We have seen that in some notations, there is no explicit representation for the business rules, as such; these notations rely on other notations for the representation of business rules. These notations include BPMN, Petri Nets and CG. Other notations such as RuleML, FCL and XTT are used purely for the representation of rules. Declarative notations such as Declare and DCR graphs offer a more "process oriented" approach where the rules represent the process model itself. With this knowledge, the modelers can make a clear decision on which notations serve their purpose for the representation of their business processes and business rules. In the next section, we will examine the tools that are available supporting these notations.

4.1.4 Tool Support (RQ4)

What kind of tool support exists for the existing modeling notations?

Tool support is one of the most vital features of a notation. The users of a notation require a tool in most cases in order to be able to make use of the notation. Modeling tools allow the user to generate models of business process and business rules. Execution tools enable the user to simulate or execute the generated models. The absence of a modeling tool renders the notation unusable in practice. Graphical modeling tools are also important especially to business experts with no technical experience, allowing them to generate models using graphs and icons while the tool generates the semantic representation behind them.

A number of tools have been implemented that support the use of DCR Graphs. In [15], DCR Workbench was used for modeling DCR graphs. The tool helps to model and visualize DCR Graphs. DCR workflow engine [16] is a multi purpose tool, which includes a process model repository, an execution host for process execution and a graphical editor for modeling and simulation. It also has a desktop client that can be used to execute instances of DCR processes. A web client also exists as well as a runtime verification tool and a model checker [16]. The Exformatics tool was also used in [19, 25]. The tool consists of 2 main features, a graphical modeling environment for modeling and simulating DCR models, and different web-services that can be used for, among other things, execution, storage, visualization and verification of DCR graphs [19, 25]. The DCR Web tool for modeling, executing and visualizing DCR Graphs has also been used in [17, 20, 21, 22 and 23]. DCR-based Transition System Miner was used in [24] to discover DCR graphs from logs.

Declare is supported by many tools ranging from modeling tools, automated discovery, conformance checking, and execution. The Declare Framework, introduced in [41] and further presented in [40], consists of three core components, Designer, Framework and Worklist. The designer is used to model the constraints in a Declare model. The Framework component is used to manage the execution of instances of the Declare model. The Worklist helps the user to access the active instances of their Declare models, which are executed by the Framework. The Declare framework has been used in [59, 66, 67, 69, 142]. Process mining tools have also been used to discover Declare models. Some of these tools are MINERful [71, 68, 66, 144], a Declarative process-mining tool that is used to discover Declare models and ProM [59], a multi faceted pluggable framework for process mining. Some of the ProM plugins that have been implemented are the Declare Checker [30, 31] and the Declare Analyzer (for conformance checking)[39], and the Declare Miner and DecMiner [50] for the discovery Declare models from execution traces [50]. Fusion Miner [52] is used to discover a combination of procedural models and Declare models from logs.

An editor for SBVR called SBVR Structured Language Editor (SBVR SLE) that can be used to define SBVR processes was proposed in [83]. The editor consists of 5 components (Figure 10): (1) package explorer that is used to manage files, (2) outlook block that shows the vocabulary concepts, (3) editing interface, (4) business rules and (5) questions. The tool is implemented with automatic error checking to correct misspelled words. It also creates links between vocabularies and between business rules and vocabulary concepts.

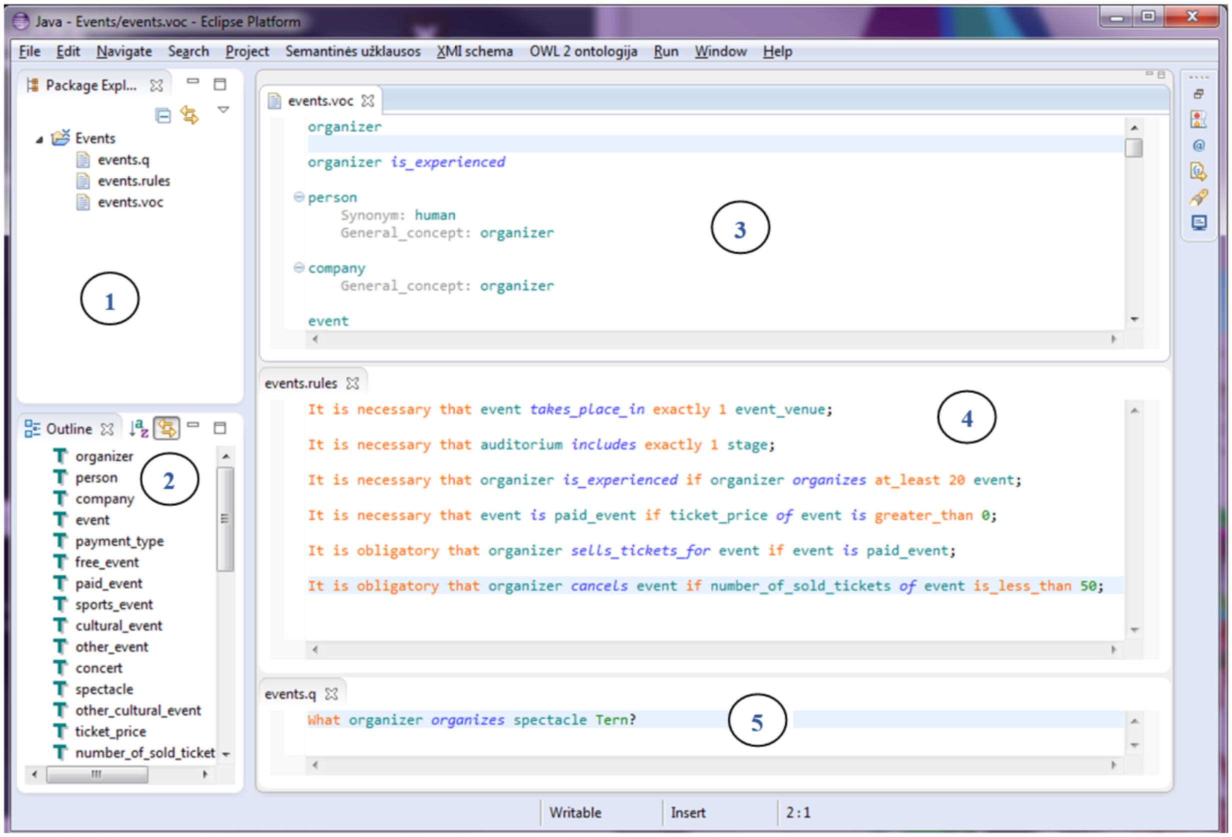


Figure 10. Environment of an SBVR Structured Language Editor (SBVR SLE) [83]

The most widely used Alloy tool is the Alloy Analyzer [199, 200, 201]. This is a tool that is used to analyze Alloy models automatically by translating the models into Boolean formulas. These formulas are then solved using SAT technology [199]. The tool consists of a frontend and a backend. The frontend transforms the input model into an intermediate representation, which is then converted into Boolean formulas by the backend [199]. [199] also proposed a tool called Kato, which accepts any Alloy model and, using declarative slicing, finds the optimal model with improved performance in terms of execution.

The Drools engine is a tool for modeling business knowledge. The rule engine of Drools allows the execution of business rules [122, 123]. The Drools engine was used in [124, 125]. RbBPDL was used in [145] to model business processes using the ECAPE notation, an extension of ECA. The tool captures five different elements: participants (person, application, entity or web service), variables (information processed or produced by the process), business activity, events and business rules. A prototype based on Drools was developed in [138] that has been used to model ECA rules. Different components of drools such as Drool's modeling environment and execution engine were extended to support the modeling and execution of event and context-aware business processes based on ECA rules [138].

Concerning the OWL notation, [126] used the Protégé Ontology Editor. This tool, developed by Stanford University is an editor for OWL. Attempto Controlled English (ACE) editor was used to write business rules in ACE, which were then mapped to the OWL notation [127]. [128] used the iWISE tool [128] to monitor and improve OWL-based business processes.

ViDRE, a distributed service oriented rule engine for RuleML was proposed in [129,130]. ViDRE is equipped with transformers that transform RuleML models into concrete rule engines. These rules are made accessible by ViDRE as web services so that business processes and applications can access them easily. ViDRE also allows the modeling of rules, which can be executed in a service-oriented manner.

[132] suggested the use of tools such as Enterprise Architect from Sparx Systems or any other open source BPMN tool for modeling or adjusting the BPMN2 models using graphical symbols and icons. In [133], PHILharmonicFlows was suggested as a tool for combining business processes with data conditions. The tool combines state-based object behavior and data driven execution of business processes [133]. It also gives the user functionalities such as creating complex UI components and prototype specification [133]. An editor for rBPMN was proposed in [139, 140]. The editor, which was built upon the Eclipse GMF framework, can be used to model any of the rBPMN rule types [140]. In [94], a plugin for the design of SBVR rules was implemented as a part of the MagicDraw CASE tool in a study that involves the integration of BPMN2 with SBVR rules.

CPN tools is an editor for CPN models and can be used to simulate and evaluate the models [150, 154]. A plugin was implemented as part of the process-mining tool ProM [152] for the mapping of activities in a process model and events in a log generated by information systems. The activities in the model are mapped to events within the log. This helps in carrying out conformance analysis and model checking. The EMBRA^{2}CE framework was used for modeling SBVR business rules in [155], which were used to constrain Petri Nets models.

Mirella Designer [156] is an XTT modeling tool that supports visual design of XTT-based models. It also provides an online verification tool. In addition, Mirella Designer can be used to transform XTT models into Prolog-based rules [156].

Oryx-HQEd Integrated Framework [157] was used to model XTT2 decision tables for BPMN. XTT2 models can be opened in Oryx and edited in HQEd via a network socket that connects the two frameworks together [157]. In [158], BPMN models in XML format were transformed into XTT2 business rules model using the BPMNtoXTT2 tool. Using a BPMN editor, such as the Eclipse modeler [158], the XML representation of a BPMN model can be exported. This model is passed as input to the BPMNtoXTT2 tool, which transforms the model into an XTT2 representation. HeaRT rule engine [158] can then be used to execute these XTT2-based decision tables [158].

Barcelona GSM design and runtime environment is a tool that provides support for the modeling and end-to-end execution of GSM BA processes. [166] proposed a framework that can be used to split and outsource GSM schemas to other parties. The framework restructures the GSM schema and applies locking protocols. This ensures that multiple occurrences of events are serialized thus preventing race conditions. It is also possible to execute the split by using an existing GSM engine to allow the reuse of existing GSM engines. The authors plan to apply the framework to CMMN. They also proposed, as part of future research to extend the framework in order to cover multi-party outsourcing as well as multiple artifact types [166].

A conceptual graph editor, called CharGer, was used in [167] to create CG models. The tool offers capabilities for modeling and generating a diagram in Conceptual Graphs Interchange Format (CGIF). Different image formats and as well a natural language representation of the model can also be exported using this tool [167]. In addition, XML representations of business rules can be exported in CGIF format [167].

In conclusion, we have examined the different notations in this section and the different tool support available for each notation. We have seen that notations such as Declare are supported by several tools. It is easier to decide which notation to choose by examining what functionalities should be supported for a given notation. In this light, this section has provided the user with knowledge of the tool support for the notations discussed so far.

4.1.5 Execution /Automation (RQ5)

To what extent do the modeling notations support automation?

Process execution is another vital aspect of business process management that allows the modeler to run the modeled processes and check the execution for violations to find aspects of the business process that require improvements. As such the availability of execution or automation mechanisms is very important for a notation. In this section, we examine the notations with regards to automation or execution support.

DCR graphs are executable models that can be automated using some of the tools mentioned earlier. DCR Workbench provides an execution environment for DCR based process models [15]. DCR Workflow Engine also has a web based client app and a windows-based desktop client that can be used for the execution of DCR graphs [16]. Another tool that can be used to execute DCR Graphs models is the Exformatics tool [19,25].

The Declare framework allows the user to create different process instances and execute them. Figure 11 shows the Declare framework's worklist component [41]. The worklist contains two active process instances (listed on the left). The second process instance is being executed. The execution environment shows that the activity medication is being executed after executing the activity examination. The activities at the center (fixation, surgery and cast) are disabled (greyed out) and therefore cannot be executed. This is because of the precedence constraint between x-ray and these activities and the fact that x-ray has not been executed. Activities x-ray, examination and medication can be executed.

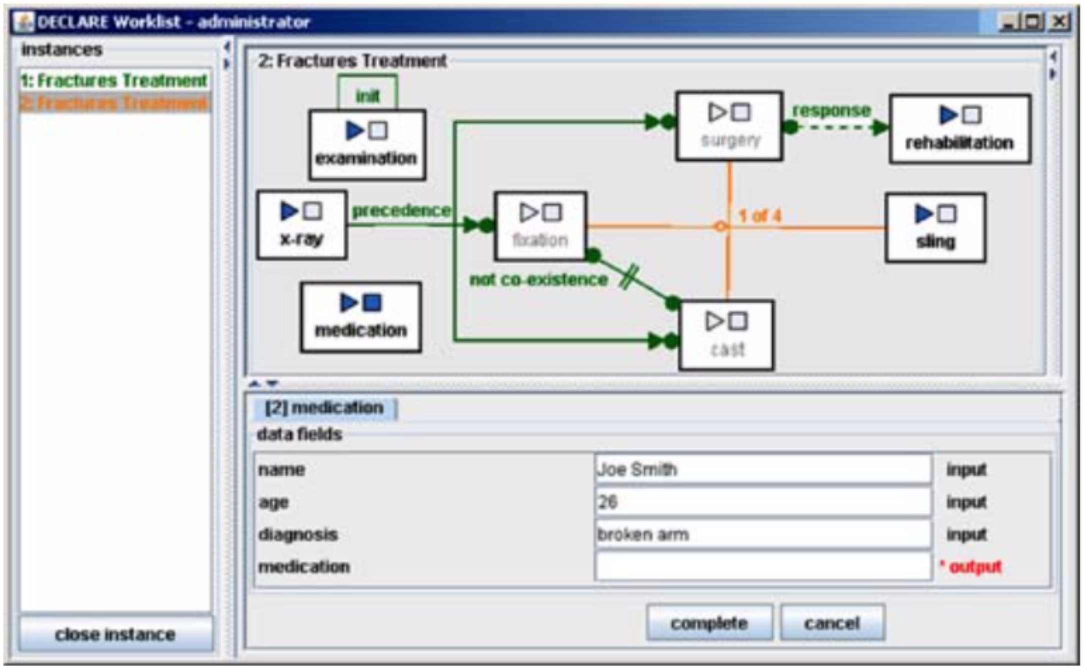


Figure 11. Declare worklist window [41]

Constraints guide the activation and the execution of activities [41]. That is, any activity that can be executed without violating the constraints is enabled. At the end of the execution of an instance, all the constraints that are not optional must be satisfied [41]. Each constraint in an instance can be in three states: satisfied (green), violated (red) or temporarily violated (yellow) [41].

SBVR models can be executed using the tool proposed in [84], which in addition to the design environment also has an execution environment that can be used to run the generated SBVR models [84]. In [95], MDBT was used to transform SBVR models into OCL and then further into executable state machine and java code. SBEAVER generates Prolog and XML schemas [96] from SBVR that can be automatically transformed into software artifacts.

Alloy models can be executed and verified using the Alloy Analyzer [199]. The tool is used to convert Alloy models into Boolean formulas that can be then executed. In [77], MP-Declare models were executed by transforming the models into Alloy formulas, which are then executed using a web tool.

Drools Framework also manages and executes business processes [122, 125]. In [125], each rule component is implemented as a Java class. These atomic classes are executed in Drools classes. The ECA business rule classes are associated with a workflow class and a rule class [125]. The Drools fusion was extended in [122] to support the management of expectations in the Event-Condition-Expectation rule format by the execution of corresponding action autonomously in accordance with the rules in case of fulfilment or violation [122].

In [126], the OWL ontology was used in the domain of healthcare to model processes, patient context, resources and rules. The Protégé system was used to model, execute and monitor the performance of an hospital workflow.

VIDRE, the tool for modeling RuleML rules can be used to execute distributed business rules [129]. The rule Engine component of VIDRE has a rule base that is responsible for storing business rules for execution [129, 130]. The data upon which the rules are executed is stored in the working memory. A pattern matcher is responsible for deciding which rules in the rule base are to be applied by checking the content of the memory [130]. Using the pattern matcher, the inference engine decides which rules are to be activated at any given point during the execution [129, 130].

In [133], it was argued that a BPMN can be executable if it is suitable to be deployed in a running BPM environment. For a model to be executable, it must satisfy the execution guidelines by conforming to the sub-classes of execution specified in the BPMN specification [133]. The execution of process models based on BPMN and its derivatives have been discussed in [134, 135, 94] but the manner in which the processes are executed and the environments were not discussed. The remaining papers that cover BPMN and BPMN extensions have also discussed process execution without going into details about how these processes are executed by the execution engine. In [157], two solutions were proposed for the execution of BPMN2 models with business rules. The first approach uses a Business Process Execution Language (BPEL) workflow executed in a BPEL workflow engine that communicates with a rule engine, HeaRT (that executes XTT2 decision tables), in such a way that the XTT2 rules are taken into consideration when executing the BPEL workflow [157]. The second approach involves the use of BPMN engines such as jBPMN that runs BPMN2 business processes and the integration of the native workflow engine with HeaRT [157, 158].

The CPN tools used in [150, 154] can be used to simulate CPN-based models. Executable CPN codes are generated in CPN ML, a high-level programming language for executing CPN models.

GSM models can be fully executed using the Barcelona modeling and execution environment [164]. By providing support for rapid model-driven development and end-to-end execution, this tool allows modeling, management and execution of

GSM models [164]. Another GSM execution engine has been introduced in [166] as a way to execute GSM models shared by different parties.

Looking at all the notations, we have seen that virtually all have some execution environment. For some of the notations, the modeling environment also supports execution and for others, the models have to be imported into the execution environment, which is either a separate engine or a component within the same process execution engine.

4.1.6 Modeling Scenario (RQ6)

In which scenarios have these notations been applied?

Looking at the areas where a notation has been applied will guide the modelers in choosing what notation best suits their needs.

DCR Graphs have been applied to numerous studies involving different types of business processes. DCR Graphs were used to model a Mortgage application [15], a cross-organizational case management system [16, 23] and to support the execution of the business process of a funding agency [17]. DCR Graphs have also been applied on the case management of hospital patients [25] as well as to generate dynamic evaluation forms [20] used a post-hoc evaluation form for a Danish arbitration court.

The most widely used declarative notation, Declare, has been used to model and analyze a wide range of business processes. For example, a log of a hospital that records cancer treatment was used to discover MP-Declare models in [27]. Logs from hospitals, banks and the academic sector have also been mined in order to verify case studies involving discovery and conformance checking of Declare models [30, 31, 37, 38, 39, 43, 45]. Declare has also been used to model clinical processes in the context of ovarian carcinoma surgery [32]. Other areas where the Declare notation has been applied include insurance claim handling [63, 53], incident and problem management cases [47, 75], and course plan verification and checking of enrollment history [143].

SBVR has also been applied in quite a number of scenarios. It has been used to model a process involving Item sales system [85, 107], in the EU-Rent process [88, 91, 102, 109], an auctioning service [89] and a car rental process [90]. In [101] three processes were modeled for case studies involving the UK taxation policy program, a bank fund transfer policy and a segment of a large legacy insurance. SBVR has also been used to model business rules for a hospital alert system [104.

116] as well as to model the business rules regulating processes of the Bank of England [117].

Alloy has been used in three different scenarios in [119]: to model a binary search tree, a red-black tree and a double-linked list. The experiments were carried out in order to improve the performance in checking Alloy models in terms of execution time. The Firewire 'tree identify' protocol model was used along with small studies in [120] in order to validate the study on debugging overconstraint in declarative models [120].

Drools' rule engine was used in [122] in a medical case study. In [123], the Iris, Balance Scale and Glass datasets were used to perform an experimental modeling using the ECA approach. The study focused on learning business rules with association rules [123]. A decision support was modeled in [124] in the domain of tourism.

In [126], the OWL ontology was used in the domain of healthcare to model processes, patient context, resources and rules. In [202], OWL was used to model the organization of a political event.

A supply-chain management case was modeled using RuleML in [129, 130] in the domain of computer manufacturing to validate the feasibility of VIDRE. RuleML was applied in the legal domain in [131].

In [134], a healthcare case study for the standardization of risk screening processes was carried out to validate the use of business rule management in BPMN2. A bookstore case study was implemented in [139] taking into account variabilities such as the customer discount, book availability, customer decision in case the book is available, decision to return the book, payment rollback and so on. These variabilities are modeled as business rules. A service-oriented healthcare management system was modeled as a proof of concept in [140] to support the feasibility of the rBPMN notation.

In [150], a case study modeling the calculation of parcel and letter costs for a post rate service using the CPN notation is presented. A collection of processes from the BIT process library was modeled in [152]. The processes include telecom, financial services and other domains.

XTT was used in [156] to model a process involving vehicle insurance products. In [157, 158], XTT2 was used in a case study involving the integration of BPMN with XTT2-based rules. The case study was carried out on a process for the determination of insurance prices. Other studies were carried out in [159] on the Polish Liability Insurance, the EUrent and the UServ Financial Services cases.

In [165], GSM was used to model a real-life process of an international tech company. This company has multiple offices in different countries. Each office can use its own variant of the process. There also exists a standard variant shared by all the offices and an office can either use its variant or the standard version. The approach proposed in [164] was validated on a supply chain solutions design enablement called the Solution Builder.

In [170], a contract service written in the SBVR notation was transformed into FCL rules in order to validate the approach for transforming SBVR models into FCL [170].

In this section, we have used the data extracted from the selected papers to identify the application domains for the available notations used to model business rules. Answering all the research questions defined in Section 3 provides the basis for the development of our classification framework.

4.2 Threats to Validity

In this section, we present the threats to the validity of our research.

a. Researcher bias

Due to the fact that only one researcher worked on the data synthesis, there might have been some bias in interpreting the data that has been extracted from the primary sources [204]. Although the study was reviewed by the other researchers, there might have been the absence of a second during the data extraction stage. As such, other researchers reviewed the extracted data to ensure that this threat is mitigated.

b. Proportion of literature between notations

During the search stage, we found and filtered the list of papers. We found out that for some notations such as Declare and SBVR, we had over 20 papers. But for other notations like FCL, we had only a few papers. Notations like Declare and SBVR may have thus been more extensively reviewed in relation to the other notations. Also, the extracted data for the notations with a low number of research papers may be inadequate in some aspects due to the restricted number of papers to extract the data from.

c. Study inclusion/exclusion bias

This threat occurs during the filtering stage of the SLR and results from generic or conflicting criteria [204]. This problem may lead to the research being too generic or the size of the literature being too large or too small. And since the filtering is used to select only the papers related to the specific subject under analysis, this may often lead to problems during data extraction like having too many papers that are irrelevant to the research topic. To mitigate this issue, we restricted the inclusion criteria to those papers that are within the topic of business process and business rules modeling. We avoided conflicts in the filtering criteria by extensively discussing and adjusting the filtering criteria [204].

d. Construction of the search string

This problem occurs when constructing the search strings. This could result in relevant papers not appearing in the search results. The search may also return too many papers that are not relevant to the research [204]. As a first measure to avoid this, we ensured that the search terms are only those terms that are part of the research questions. We also included synonyms of search terms. As a precautionary measure, we also discussed those search terms and subsequently, the search results in order to ensure that bias of this nature is avoided [204].

e. Selection of Digital Libraries

This may occur when selecting the libraries to use for the search. The search engine could be too generic or too specific. Some digital libraries also contain an insufficient number of papers [204]. We selected two of the biggest most widely used digital libraries, i.e., Scopus and Web of Science. A discussion was also carried out in order to make this decision. With this, we were able to curtail if not eliminate this threat.

f. Coverage of research questions

This occurs when the research questions do not fulfill the goal of the study. When a research topic is not adequately decomposed into research questions or when the goal is too generic, this threat arises [204]. To avoid this, we formulated the main research question to define the research goal. We then discussed and decomposed the question into specific sub-questions.

5. Business Rule Evaluation Framework

In the previous section, we tried to answer the research questions defined in this literature review. First, we examined the different existing notations that can be used for modeling business rules. We then looked at the expressive power of these modeling notations, i.e., the aspects of a business process that they can capture such as data, resources, and time. We also looked at how the business rules can be represented and integrated into business process models. Furthermore, we examined the tools available supporting these notations such as modeling and analysis tools. We also looked at the existence of execution engines supporting the enforcement of business rules during the execution of a business process. Finally, we analyzed the different scenarios where these notations have been applied.

To tie it all up, we designed a framework for classifying business rules based on different aspects. A brief overview of the framework is given in Table 8. The first and the third research questions of our study give an overview of all the notations that are available for modeling business rules and how these notations are integrated into business process models. The types of notations and their modeling approaches thus cover the first three aspects of the framework, giving the user insights into the different business rule modeling notations that exist. The second research question covers the expressivity of the modeling notations. Therefore, in the proposed framework, we looked at the representation of time, deontic logic as well as data and resources. In terms of tool support as discussed in the fourth research question, we covered the two most important tool types that should support a modeling notation, which are modeling and analysis tools. This covers the reasoning aspect of the framework, which consists of model verification and online/offline analysis. The tool support was further discussed in the fifth research question where we looked at how the different notations support automation and the tools that are used to execute the models. In the final research question, we discussed the different scenarios where the notations are used and this covers the last aspect of the framework.

The framework allows the modelers to examine business rule modeling notations based on the criteria that we just discussed. The different criteria are discussed in more detail in the next subsection. Table 8 provides the main components of the proposed framework. These components are further broken down into more specific components. At the end of this section, we show how the framework has been applied to classify the studies selected in this literature review.

1. Language / Notation
2. Notation type
3. Modeling paradigm
4. Expressivity
5. Syntax/ Semantics
6. Tool Support
7. Reasoning
8. Modeling Scenario

Table 8. The general components of the framework

5.1 Language / Notation

Here, we list the notations that have been identified in the literature. This gives the user an overview of the available notations that can be used to model business rules. It lays the ground for identifying the different properties or functionalities of that particular notation.

5.2 Notation Type

We also identified the different modeling approaches that have been used to capture business rules. Here, we identify whether the business rules are represented implicitly (such as in BPMN [94]), explicitly modeled as a declarative business process model (such as in Declare [40]), or as using a notation that needs to be integrated into a separate process model such as XTT [156] and RuleML [130].

5.3 Modeling Paradigm

In business process modeling, we identified the two modeling paradigms: imperative and declarative. Declarative modeling uses business rules to explicitly constrain the business process. Notations such as Declare and DCR Graphs [15] follow a declarative approach. Imperative modeling focuses on modeling the business process where the rules are either implicitly modeled or stored as a separate model. Petri Nets [151] and BPMN [94] are imperative modeling notations.

5.4 Expressivity

Here, we explored how expressive the notation is. We looked at the possibility of expressing some aspects of a process using the notation. The different aspects include: time, deontic aspects, data and resources.

5.4.1 Time

With respect to the expression of time we looked at how time can be modeled. To express time, we can express time as qualitative or metric time. Qualitative time is expressed in terms of the relative position of events. For instance, in the case of laundry service, the washing of clothes comes before ironing. We understand ironing happens after washing, but the time here is relative as opposed to metric. Metric time on the other hand expresses time using the concept of timestamp. For instance, acceptance of applications starts on Monday at 11:00 am. The time expressed can either be a timestamp or time interval in the form of a duration. The original version of Declare for instance, does not support metric time, but Timed-Declare [76] supports it. The ability to express time is an important feature in a modeling notation especially for time critical processes where the timing of an event or process component affects the outcome of the process. As a result we included time as an aspect of our framework.

5.4.2 Deontic aspects

Deontic logic deals with the possibility of modeling permissions, obligations, prohibitions and negations. Deontic logic was also included in the framework as it permits the representation of these four components. SBVR supports deontic logic representation. Process compliance to certain policies within the organization or policies that are designed by the authority can be represented using deontic logic. This helps the businesses to stay compliant during the execution of their processes.

5.4.3 Data

Data is another aspect of the business process we considered in our framework. Data is an important part of most businesses. As such the capability of representing data as part of the business process is of great value for a notation. This is true especially for businesses that are data-centric.

5.4.4 Resources

Here, we focus on the possibility of modeling resources in the business rule notations. Resources may include, e.g., actors, devices and money. In some cases, resources are represented as data. Resources are the backbone of any business and the pillars upon which the business stands. A modeling notation that provides not just the representation of activities and rules but also the cost of the activities and the person or machine performing the activities can be said to have great expressive capabilities.

5.5 Syntax/ Semantics

5.5.1 Formal versus informal semantics

Semantics deals with the meaning of the business rules. Formal semantics provides a formal description of the rule meaning. Informal semantics describe rules using an informal description like plain natural language.

5.5.2 Formal versus informal syntax

Syntax represents the rules that govern how the different components are arranged. This can also be either formal or informal.

5.6 Tool Support

This section deals with the tool support available for each notation.

5.6.1 Modeling environment

A modeling environment allows the user to design their business rules using a graphical user interface. An alternative to that is to represent the models on paper. Since nowadays it is hardly feasible to think about a modeling notation that cannot be used to create digital documents, the availability of a modeling environment is a very important aspect

.

5.6.2 Execution environment

An execution environment provides a mechanism to run a business process. Some notations can in fact be executed and as such, for these notations, an execution environment is required. For businesses that require more than just graphical representation, an execution mechanism for the notation they choose is a must. For this reason, we have included this as part of our framework.

5.7 Reasoning

5.7.1 Consistency

Checking business rule models for consistency is another aspect that is very important for a business rules modeling notation. This helps the modeler to find loopholes and improvements in their business processes and make adjustments. It involves the design time checking of a model against a set of business rules in order to ensure that the business rules are satisfiable.

5.7.2 Online / Offline analysis

Online analysis involves the checking of process executions against a set of business rules. This is also called compliance monitoring. Offline analysis or conformance checking on the other hand involves checking of completed traces with respect to a set of business rules. A mechanism such as this is very important as the lack of this mechanism could lead to errors that are detrimental to the process or the entire business.

In Table 10, we present the framework applied to the notations that we have extracted in the previous sections outlining all of the above components. We have also included references to the notations where the user can find details about the notations and how these different notations have been applied.

In order to use the framework, the modelers need to identify the notation they would like to apply. If for instance the user wants to use DCR Graphs, they can use the framework to identify what can or cannot be modeled with DCR Graphs. The features of DCR Graphs as described in the framework can be identified as follows: The notation is used to design business rules in the form of a business process model. DCR Graphs are based on a declarative approach to business process modeling. They do not support the modeling of data nor deontic logic. However, resources can be represented. It has formal semantics. DCR Graphs also offer modeling and execution environments with support for consistency checking and online/offline analysis. This notation has been used in a variety of cases ranging from mortgage application processes to patient management in a hospital. References related to DCR Graphs are also specified.

Another case is where the user does not in fact have a notation in mind. In this case, they can use the framework to identify which notation is appropriate for their case using the features described starting from the second column (**Notation type**) and discarding any row(s) that fails to meet their criteria. At the end, the user can identify those rows that are left and come back to check which notations are the most appropriate for their use case.

1. Language / Notation	2. Notation type	3. Modeling paradigm	4. Expressivity				5. Syntax/ Semantics	6. Tool Support		7. Reasoning			
			4.1 Time (Qualitative/ metric)	4.2 Deontic aspects	4.3 Data	4.4 Resources	5.1 Formal / Informal	6.1 Modelling environment	6.2 Execution environment	7.1 Online/ offline analysis	7.2 Consistency checking	8. Modeling Scenario	Ref
DCR Graphs			Both	No	No	Yes	Formal	Yes	Yes	Yes	Yes	Mortgage application. A cross-organizational case management system. The business process of a funding agency. The case management of hospital patients. The Dreyer Foundation grants for lawyers and architects process. Arbitration court evaluation forms	15,16,17, 19, 20, 21, 22, 23, 24, 25
Declare		Declarative	Qualitative / M	No	Possible	Possible	Formal	Yes	Yes	Yes	Yes	A book purchasing process. Data from logs obtained from multiple Dutch municipalities. Data obtained from real-life as well as synthetic log of a hospital. a clinical scenario in the context of ovarian carcinoma surgery. A log obtained from financial domain. Logs from the urology department of a hospital. The process of acknowledging an unborn child in multiple municipalities. Logs involving cancer treatment process. Insurance claim handling and incident and problem management. Monitoring of vessel behavior. A real-life process involving the diagnosis and treatment of patients with arm fracture. Road traffic fines management process. Road traffic management log. Order-To-Cash process	26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 39, 40, 41, 42, 43, 44, 45, 47, 48, 49, 50, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 78, 79, 80, 81, 82, 142, 143, 144, 145, 146, 148
MP-Declare	Busines Process		Metric	No	Yes	Yes	Formal	Yes	Yes	Yes	Yes	The logs from a hospital that records cancer treatment from a hospital. Data obtained from three different organizations, a bank, a financial institution and an academic hospital. Logs from an academic hospital a local police office and five municipalities	27, 37, 38, 46, 51, 77
GSM			Qualitative	No	Yes	Yes	Formal	Yes	Yes	No	No	A real-life process of an international tech company	164, 165, 166
Drools			Metric	No	Possible	Possible	Formal	Yes	Yes	Yes	Yes	A medical case study in order to validate the handling of expectations in Drools. The Iris, Balance Scale and Glass datasets were used to perform an experimental modeling. A decision support in the domain of tourism. A construction process	122, 123, 124, 125
ECA			Metric	No	Possible	No	Formal	Yes	Yes	No	No	Electronic brokerage process	146, 145, 183
Alloy			Metric	No	Yes	No	Formal	Yes	Yes	Yes	Yes	To model a binary search tree, red-black tree and double-linked list. Different varieties of experiments along with small studies	119, 120, 121
Petri Nets			Qualitative	No	No	No	Formal	Yes	Yes	Yes	Yes	Modeling the calculation of parcel and letter cost for a post rate service. A collection of processes from the BIT process library including telecom, financial services and other domains	150, 151, 152, 153, 154, 155
BPMN		Imperative	Metric	No	Yes	Yes	Formal	Yes	Yes	Yes	Possible	A hospital case study involving the standardization of risk screening processes. An industrial process. A bookstore process case study. A service-oriented healthcare management system. A number of selected instances of different business processes including the EURent case, Polish Library Insurance case and so on	132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 94
Conceptual Graphs			Qualitative	No	Yes	Yes	Formal	Yes	Possible	No	No	-	167, 168
BPEL			Qualitative	No	Yes	No	Formal	Yes	Yes	No	No	Foundry business	160, 64
FCL			Qualitative	Yes	No	No	Formal	Yes	No	No	No	A sample contract service	169, 170
SBVR	Business Rules		Qualitative / M	Yes	Possible	Possible	Informal	Yes	Yes	Yes	Yes	Item sales system. The EU-Rent process. An auctioning service. A case study involving car rental. The taxation policy, fund transfer and a large legacy insurance. Hospital alert systems for the validation of drug prescriptions. The business rules for the Bank of England	83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118
XTT			No	No	No	No	Informal	Yes	No	No	Yes	A process involving vehicle insurance products. A process for the determination of insurance price. Polish Liability Insurance, EUrent and the UServ Financial Services	156, 157, 158, 159, 160
RuleML			Metric	Yes	Possible	Possible	Formal	Yes	Yes	No	No	A supply-chain management case	129, 130, 131
OWL			Metric	No	Yes	Yes	Formal	Yes	Yes	Yes	Yes	In the domain of healthcare to model processes, patient context, resources and rules. Modeling the organization of a political event including resources, persons, organization and time	126, 127, 128

Table 10 The Proposed Framework

6. Conclusion

We have examined different notations to model business rules. We carried out an SLR to examine the notations available in the literature. We have carried out an evaluation of the existing business rule modeling notations based on a number of criteria such as type of notation, expressivity, tool support, application domain. We presented the results of our findings along with the steps taken to achieve the results.

The research questions driving the SLR were used to create a framework that can be used to classify the existing modeling notations. The framework is composed of the following features: language, notation type, modeling paradigm, expressivity, syntax and semantics, tool support, reasoning and modeling scenario. The language is the identifier or name that is used to identify the language or notation.

Our study does not include certain aspects of modeling such as implicit and explicit time representation. Implicit time representation is the representation of time using adjectives while explicit time representation is the representation of time using explicit terms such as One hour, 11pm and so on. Another aspect of time representation we did not include is the point-based/interval-based representation. Interval-based time representation deals with the representation of time with respect to the passing of a duration. For instance, after 30 minutes. Point-based time representation deals with the representation of time w.r.t a particular point in time e.g. 11:00pm. Other aspects not considered in this study include usage context (industrial/academic, generic/domain-specific), documentation, statistical analysis tools and natural language support for model specification among other things. As future study, we will expand the framework to cover other aspects of business rule modeling that we have mentioned here along with others that we have not included.

7. References

1. Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering. Engineering. 2, 1051 (2007).

2. Fink, A.: Conducting research literature reviews: from the Internet to paper. (2010).

3. Levy, Y., Ellis, T.J.: A systems approach to conduct an effective literature review in support of information systems research. Informing Sci. 9, 181–211 (2006).

4. Okoli, C.: A guide to conducting a standalone systematic literature review. Commun. Assoc. Inf. Syst. 37, 879–910 (2015).

5. Webster, J., Watson, R.T.: Analyzing the past to prepare the future. MIS Q. 26, xiii–xxiii (2002).

6. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele Univ. 33, 28 (2004).

7. Brereton, P., Kitchenham, B., Budgen, D.: Lessons from applying the systematic literature review process within the software engineering domain. … Syst. Softw. 80, 571–583 (2007).

8. Mistiaen, P., Francke, A.L., Poot, E.: Interventions aimed at reducing problems in adult patients discharged from hospital to home: a systematic meta-review. *BMC Health Serv*. Res. 7, 47 (2007).

9. Randolph, J.J.: A Guide to Writing the Dissertation Literature Review. Pract. Assessment, Res. Eval. 14, 1–13 (2009).

10. F. Rosenberg and S. Dustdar. Business Rules Integration in BPEL – A Service-Oriented Approach. *Seventh IEEE International Conference on E-Commerce Technology*, July, 2005.

11. The Business Rules Group. Defining Business Rules – What Are They Really? http://www.businessrulesgroup.org/first paper/br01c0.htm, July 2000.

12. A. Charfi, M. Mezini, Hybrid Web Service Composition: Business Processes Meet Business Rules. *In Proceeding Second International Conference on Service-Oriented Computing*, November, 2004.

13. B. von Halle and L. Goldberg. What is the Decision Model? - An excerpt from The Decision Model: A Business Logic Framework Linking Business and Technology. http://www.ittoday.info/ITPerformanceImprovement/Articles/2011-03VonHalleGoldberg.html, March 2011.

14. A. Briggs, M. Sculpher, K. Claxton Decision Modelling for Health Economic Evaluation

15 Debois, S., Hildebrandt, T. and Slaats, T. (2015). Concurrency and Asynchrony in Declarative Workflows. [97]

16 Thomas Hildebrandt, T. (2011). Designing a cross-organizational case management system using dynamic condition response graphs. [online] Citeseerx.ist.psu.edu. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.701.2585. [98]

17 Heuck, E., Hildebrandt, T., Kiærulff Lerche, R., Marquard, M., Normann, H., Iven Strømsted, R. and Weber, B. (2017). Digitalising the General Data Protection Regulation with Dynamic Condition Response Graphs. [online] Core.ac.uk. Available at: https://core.ac.uk/display/144131032. [96]

18 Xu Chu, P. (2019). Discovering Denial Constraints. [online] Citeseerx.ist.psu.edu. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.406.1894 [Accessed 12 Aug. 2019]. [95]

19 Hildebrandt, T., Marquard, M., Mukkamala, R. and Slaats, T. (2013). Dynamic Condition Response Graphs for Trustworthy Adaptive Case Management. [99]

20 Strømsted, R., A. Lopez, H., Debois, S. and Marquard, M. (2018). Dynamic Evaluation Forms using Declarative Modeling. *IT-University of Copenhagen*. [online] Pure.itu.dk. Available at: https://pure.itu.dk/portal/en/publications/dynamic-evaluation-forms-using-declarative-modeling(239bb72a-04b3-45fa-aa9a-44f42969e58a).html [Accessed 12 Aug. 2019]. [100]

21 Hildebrandt, T., Mukkamala, R. and Slaats, T. (2012). Nested Dynamic Condition Response Graphs. [101]

22 Debois, Søren & Hildebrandt, Thomas & Slaats, Tijs. (2017). Replication, refinement & reachability: complexity in dynamic condition-response graphs. *Acta Informatica*. 55. 10.1007/s00236-017-0303-8. 105

23 Hildebrandt, T., Mukkamala, R. and Slaats, T. (2011). Safe Distribution of Declarative Processes. [102]

24 Debois, S. and Slaats, T. (2015). The Analysis of a Real Life Declarative Process. [online] Semanticscholar.org. Available at: https://www.semanticscholar.org/paper/The-Analysis-of-a-Real-Life-Declarative-

Process-Debois-Slaats/13f27ee6a0d897c31db4cb172b784d0f4668a651 [Accessed 12 Aug. 2019]. [104]

25 Mukkamala, R., Hildebrandt, T. and Slaats, T. (2013). Towards Trustworthy Adaptive Case Management with Dynamic Condition Response Graphs. [online] Core.ac.uk. Available at: https://core.ac.uk/display/50527643 [Accessed 12 Aug. 2019]. [103]

26 Pesic, Maja & Aalst, Wil M. P.. (2006). A Declarative Approach for Flexible Business Processes Management. 4103. 169-180. 10.1007/11837862_18. 86

27 Sturm, Christian & Schönig, Stefan & Jablonski, Stefan. (2018). A MapReduce Approach for Mining Multi-Perspective Declarative Process Models. 585-595. 10.5220/0006710305850595. 161

28 Hanser, Michael & Di Ciccio, Claudio & Mendling, Jan. (2016). A Novel Framework for Visualizing Declarative Process Models. 106

29 Montali, Marco & Torroni, Paolo & Chesani, Federico & Mello, Paola & Alberti, Marco & Lamma, Evelina. (2010). Abductive Logic Programming as an Effective Technology for the Static Verification of Declarative Business Processes. Fundam. Inform.. 102. 325-361. 10.3233/FI-2010-310. 87

30 de Leoni, Massimiliano & Maggi, Fabrizio & Aalst, Wil M. P.. (2012). Aligning event logs and declarative process models for conformance checking. 174

31 de Leoni, Massimiliano & Maggi, Fabrizio & Aalst, Wil M. P.. (2014). An Alignment-based Framework to Check the Conformance of Declarative Process Models and to Preprocess Event-Log Data. *Information Systems.* 47. 10.1016/j.is.2013.12.005. 107

32 Jimenez Ramirez, Andres & Barba, Irene & Reichert, Manfred & Weber, Barbara & Del Valle, Carmelo. (2018). Clinical Processes - The Killer Application for Constraint-Based Process Interactions?. 10.1007/978-3-319-91563-0_23. 219

33 M. de Carvalho, Renata & Mili, Hafedh & Gonzalez-Huerta, Javier & Boubaker, Anis & Leshob, Abderrahmane. (2016). Comparing ConDec to CMMN: Towards a Common Language for Flexible Processes. 10.5220/0005688002330240. 92

34 Marrella, Andrea & De Giacomo, Giuseppe & Sardina, Sebastian & Maggi, Fabrizio. (2016). Computing Trace Alignment against Declarative Process Models through Planning. 112

35 Schunselaar, Dennis & Maggi, Fabrizio & Sidorova, Natalia & Aalst, Wil M. P.. (2012). Configurable Declare: Designing Customisable Flexible Process Models. 20-37. 10.1007/978-3-642-33606-5_3. 113

36 Borrego, Diana & Barba, Irene. (2014). Conformance checking and diagnosis for declarative business process models in data-aware scenarios. *Expert Systems with Applications*. 41. 5340–5352. 10.1016/j.eswa.2014.03.010. 20

37 Burattin, Andrea & Maggi, Fabrizio & Sperduti, Alessandro. (2015). Conformance Checking Based on Multi-Perspective Declarative Process Models. *Expert Systems with Applications*. 65. 10.1016/j.eswa.2016.08.040. 164

38 Leno, Volodymyr & Dumas, Marlon & Maggi, Fabrizio. (2018). Correlating Activation and Target Conditions in Data-Aware Declarative Process Discovery: *16th International Conference, BPM 2018, Sydney, NSW, Australia, September 9–14, 2018, Proceedings*. 10.1007/978-3-319-98648-7_11. 165

39 Rovani, Marcella & Maggi, Fabrizio & de Leoni, Massimiliano & Aalst, Wil M. P.. (2015). Declarative process mining in healthcare. *Expert Systems with Applications*. 42. 9236-9251. 10.1016/j.eswa.2015.07.040. 114

40 Aalst, Wil M. P. & Pesic, M & Schonenberg, Helen. (2009). Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development*. 23. 99-113. 10.1007/s00450-009-0057-9. 116

41 Pesic, Maja & Schonenberg, Helen & Aalst, Wil M. P.. (2007). DECLARE: Full support for loosely-structured processes. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*. 287-287. 10.1109/EDOC.2007.14. 117

42 Cimitile, Marta & Maggi, Fabrizio & Bernardi, Mario. (2014). Discovering Cross-Organizational Business Rules from the Cloud. 10.1109/CIDM.2014.7008694. 118

43 Maggi, Fabrizio & Dumas, Marlon & García-Bañuelos, Luciano & Montali, Marco. (2013). Discovering Data-Aware Declarative Process Models from Event Logs. 81-96. 10.1007/978-3-642-40176-3_8. 119

44 De Smedt, Johannes & Weerdt, Jochen & Serral, Estefanía & Vanthienen, Jan. (2018). Discovering Hidden Dependencies in Constraint-Based Declarative

Process Models for Improving Understandability. *Information Systems*. 10.1016/j.is.2018.01.001. 120

45 Maggi, Fabrizio. (2014). Discovering Metric Temporal Business Constraints from Event Logs. 10.1007/978-3-319-11370-8_19. 220

46 Schönig, Stefan & Di Ciccio, Claudio & Maggi, Fabrizio & Mendling, Jan. (2016). Discovery of Multi-Perspective Declarative Process Models. 9936. 10.1007/978-3-319-46295-0_6. 121

47 Bernardi, Mario & Cimitile, Marta & Di Francescomarino, Chiara & Maggi, Fabrizio. (2016). Do Activity Lifecycles Affect the Validity of a Business Rule in a Business Process?. *Information Systems*. 62. 10.1016/j.is.2016.06.002. 122

48 Maggi, Fabrizio & R.P., Jagadeesh Chandra Bose & Aalst, Wil M. P.. (2012). Efficient Discovery of Understandable Declarative Process Models from Event Logs. *Chemical Engineering & Technology - CHEM ENG TECHNOL*. ??-??. 10.1007/978-3-642-31095-9_18. 123

49 Di Ciccio, Claudio & Maggi, Fabrizio & Montali, Marco & Mendling, Jan. (2015). Ensuring Model Consistency in Declarative Process Discovery. 9253. 10.1007/978-3-319-23063-4_9. 124

50 Chesani, Federico & Lamma, Evelina & Mello, Paola & Montali, Marco & Riguzzi, Fabrizio & Storari, Sergio. (2009). Exploiting Inductive Logic Programming Techniques for Declarative Process Mining. *T. Petri Nets and Other Models of Concurrency*. 2. 278-295. 10.1007/978-3-642-00899-3_16. 89

51 Schönig, Stefan & Sturm, Christian & Fichtner, Myriel. (2019). Full Support for Efficiently Mining Multi-Perspective Declarative Constraints from Process Logs. Information (Switzerland). 2019. 10.3390/info10010029. 166

52 De Smedt, Johannes & Weerdt, Jochen & Vanthienen, Jan. (2015). Fusion Miner: Process Discovery for Mixed-Paradigm Models. *Decision Support Systems*. 77. 10.1016/j.dss.2015.06.002. 125

53 Burattin, Andrea & Cimitile, Marta & Maggi, Fabrizio. (2014). Lights, Camera, Action! Business Process Movies for Online Process Discovery. *Lecture Notes in Business Information Processing*. 202. 10.1007/978-3-319-15895-2_34. 127

54 Westergaard, M & Maggi, Fabrizio. (2012). Looking Into the Future Using Timed Automata to Provide A Priori Advice about Timed Declarative Process Models. 221

55 Haisjackl, Cornelia & Zugal, Stefan & Soffer, Pnina & Hadar, Irit & Reichert, Manfred & Pinggera, Jakob & Weber, Barbara. (2015). Making Sense of Declarative Process Models: Common Strategies and Typical Pitfalls. *Lecture Notes in Business Information Processing*. 147. 10.1007/978-3-642-38484-4_2. 90

56 Schunselaar, Dennis & Slaats, Tijs & Maggi, Fabrizio & A. Reijers, Hajo & Aalst, Wil M. P.. (2018). Mining Hybrid Business Process Models: A Quest for Better Precision. 10.1007/978-3-319-93931-5_14. 128

57 De Smedt, Johannes & Weerdt, Jochen & Vanthienen, Jan & Poels, Geert. (2015). Mixed-Paradigm Process Modeling with Intertwined State Spaces. *Business & Information Systems Engineering*. 58. 10.1007/s12599-015-0416-y. 129

58 Santos, Eduardo & Francisco, Rosemary & Vieira, Agnelo & Rocha Loures, Eduardo & Antonio Busetti de Busetti, Marco. (2011). Modeling Business Rules for Supervisory Control of Process-Aware Information Systems. 447-458. 10.1007/978-3-642-28115-0_42. 36

59 Maggi, Fabrizio & Montali, Marco & Westergaard, Michael & Aalst, Wil M. P.. (2011). Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata. *Control Engineering Practice - CONTROL ENG PRACTICE*. 6896. 132-147. 10.1007/978-3-642-23059-2_13. 130

60 Montali, Marco & Maggi, Fabrizio & Chesani, Federico & Mello, Paola & Aalst, Wil M. P.. (2013). Monitoring Business Constraints with the Event Calculus. *ACM Transactions on Intelligent Systems and Technology*. 5. 10.1145/2542182.2542199. 91

61 De Masellis, Riccardo & Maggi, Fabrizio & Montali, Marco. (2014). Monitoring Data-Aware Business Constraints with Finite State Automata. ACM International Conference Proceeding Series. 10.1145/2600821.2600835. 131

62 Di Ciccio, Claudio & Maggi, Fabrizio & Montali, Marco & Mendling, Jan. (2018). On the Relevance of a Business Constraint to an Event Log. *Information Systems*. 10.1016/j.is.2018.01.011. 132

63 Burattin, Andrea & Cimitile, Marta & Maggi, Fabrizio & Sperduti, Alessandro. (2015). Online Discovery of Declarative Process Models from Event Streams. *IEEE Transactions on Services Computing*. 8. 10.1109/TSC.2015.2459703. 133

64 Maggi, Fabrizio & Burattin, Andrea & Cimitile, Marta & Sperduti, Alessandro. (2013). Online Process Discovery to Detect Concept Drifts in LTL-Based Declarative Process Models. 8185. 10.1007/978-3-642-41030-7_7. 134

65 Maggi, Fabrizio & Di Ciccio, Claudio & Di Francescomarino, Chiara & Kala, Taavi. (2017). Parallel Algorithms for the Automated Discovery of Declarative Process Models. *Information Systems*. 10.1016/j.is.2017.12.002. 135

66 Di Ciccio, Claudio & Maggi, Fabrizio & Montali, Marco & Mendling, Jan. (2016). Resolving inconsistencies and redundancies in declarative process models. *Information Systems*. 64. 10.1016/j.is.2016.09.005. 109

67 Maggi, Fabrizio & Westergaard, Michael & Montali, Marco & Aalst, Wil M. P.. (2012). Runtime Verification of LTL-Based Declarative Process Models. 7186. 131-146. 10.1007/978-3-642-29860-8. 110

68 Maggi, Fabrizio & Montali, Marco & Di Ciccio, Claudio & Mendling, Jan. (2016). Semantical Vacuity Detection in Declarative Process Mining. 158-175. 10.1007/978-3-319-45348-4_10. 111

69 Di Ciccio, Claudio & Mecella, Massimo & Mendling, Jan. (2015). The Effect of Noise on Mined Declarative Constraints. *Lecture Notes in Business Information Processing*. 203. 1-24. 10.1007/978-3-662-46436-6_1. 136

70 Mertens, Steven & Gailly, Frederik & Poels, Geert. (2017). Towards a decision-aware declarative process modeling language for knowledge-intensive processes. *Expert Systems with Applications*. 87. 316–334. 10.1016/j.eswa.2017.06.024. 108

71 Back C.O., Debois S., Slaats T. (2018) Towards an Empirical Evaluation of Imperative and Declarative Process Mining. In: Woo C., Lu J., Li Z., Ling T., Li G., Lee M. (eds) Advances in Conceptual Modeling. ER 2018. *Lecture Notes in Computer Science*, vol 11158. Springer, Cham 137

72 Cornelia, Haisjackl & Barba, Irene & Zugal, Stefan & Soffer, Pnina & Hadar, Irit & Reichert, Manfred & Pinggera, Jakob & Weber, Barbara. (2014). Understanding Declare Models: Strategies, Pitfalls, Empirical Results. *Software and Systems Modeling*. 15. 10.1007/s10270-014-0435-z. 233

73 Maggi, Fabrizio & Mooij, Arjan & Aalst, Wil M. P.. (2011). User-guided discovery of declarative process models. Chromatographia. 192-199. 10.1109/CIDM.2011.5949297. 234

74 Bernardi, Mario & Cimitile, Marta & Di Lucca, Giuseppe & Maggi, Fabrizio. (2012). Using Declarative Workflow Languages to Develop Process-

Centric Web Applications. *Proceedings of the 2012 IEEE 16th International Enterprise Distributed Object Computing Conference Workshops, EDOCW 2012*. 56-65. 10.1109/EDOCW.2012.17. 237

75 Bernardi, Mario & Cimitile, Marta & Di Francescomarino, Chiara & Maggi, Fabrizio. (2014). Using Discriminative Rule Mining to Discover Declarative Process Models with Non-Atomic Activities. 8620. 10.1007/978-3-319-09870-8_21. 238

76 Maggi, Fabrizio & Westergaard, Michael. (2014). Using Timed Automata for a Priori Warnings and Planning for Timed Declarative Process Models. International Journal of Cooperative Information Systems. 23. 10.1142/S0218843014400036. 239

77 Schönig, Stefan & Ackermann, Lars & Schützenmeier, Nicolai & Petter, Sebastian & Jablonski, Stefan. (2018). Execution of Multi-Perspective Declarative Process Models. 58

78 De Masellis, Riccardo & Di Francescomarino, Chiara & Ghidini, Chiara & Laponin, Arne & Maggi, Fabrizio. (2017). Rule Propagation: Adapting Procedural Process Models to Declarative Business Rules. 165-174. 10.1109/EDOC.2017.30. 43

79 De Masellis, Riccardo & Di Francescomarino, Chiara & Ghidini, Chiara & Maggi, Fabrizio. (2016). Declarative Process Models: Different Ways to Be Hierarchical. 9936. 10.1007/978-3-319-46295-0_7. 115

80 Bryl, Volha & Mello, Paola & Montali, Marco & Torroni, Paolo & Zannone, Nicola. (2008). B-Tropos: Agent-Oriented Requirements Engineering Meets Computational logic for Declarative Business Process Modeling and Verification. 157-176. 10.1007/978-3-540-88833-8_9. 63

81 Chesani, Federico & Mello, Paola & Montali, Marco & Riguzzi, Fabrizio & Sebastianis, Maurizio & Storari, Sergio. (2008). Checking Compliance of Execution Traces to Business Rules. 17. 134-145. 10.1007/978-3-642-00328-8_13. 88

82 Li, Guangming & M. de Carvalho, Renata & Aalst, Wil M. P.. (2017). Automatic Discovery of Object-Centric Behavioral Constraint Models. 43-58. 10.1007/978-3-319-59336-4_4. 167

83 Šukys, Algirdas & Ablonskis, Linas & Nemuraite, Lina & Paradauskas, Bronius. (2016). A grammar for ADVANCED SBVR editor. *Information Technology And Control*. 45. 10.5755/j01.itc.45.1.9219.

84 De Roover, Willem & Caron, Filip & Vanthienen, Jan. (2011). A Prototype Tool for the Event-Driven Enforcement of SBVR Business Rules. *Lecture Notes in Business Information Processing*. 99. 446-457. 10.1007/978-3-642-28108-2_43.

85 Malik, Saleem & Bajwa, Imran. (2012). A Rule Based Approach for Business Rule Generation from Business Process Models. 10.1007/978-3-642-32689-9_8.

86 Mitra, S., & Chittimalli, P.K. (2017). A Systematic Review of Methods for Consistency Checking in SBVR-based Business Rules. DIAS/EDUDM@ISEC.

87 Anand, Kritika & Chittimalli, Pavan & Naik, Ravindra. (2018). An Automated Detection of Inconsistencies in SBVR-based Business Rules Using Many-sorted Logic. 10.1007/978-3-319-73305-0_6.

88 Steen, Bas & Ferreira Pires, Luis & Iacob, Maria-Eugenia. (2010). Automatic Generation of Optimal Business Processes from Business Rules. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*. 117-126. 10.1109/EDOCW.2010.40.

89 Malik, Saleem & Bajwa, Imran. (2012). Back to Origin: Transformation of Business Process Models to Business Rules. *Lecture Notes in Business Information Processing*. 132. 10.1007/978-3-642-36285-9_61.

90 Chittimalli, Pavan & Anand, Kritika. (2016). Domain-independent method of detecting inconsistencies in SBVR-based business rules. 9-16. 10.1145/2975941.2975943.

91 Skersys, Tomas & Tutkute, Lina & Butleris, Rimantas & Butkienė, Rita. (2012). Extending BPMN Business Process Model with SBVR Business Vocabulary and Rules. *Information technology and control*. 41. 10.5755/j01.itc.41.4.2013.

92 Lévy, François & Nazarenko, Adeline. (2013). Formalization of Natural Language Regulations through SBVR Structured English. 8035. 10.1007/978-3-642-39617-5_5.

93 Cheng, Ran & Sadiq, Shazia & Indulska, Marta. (2011). Framework for business process and rule integration: A case of BPMN and SBVR. *Lecture Notes in Business Information Processing*. 87. 13-24. 10.1007/978-3-642-21863-7_2.

94 Butleris, Rimantas & Mickeviciute, Egle & Nemuraite, Lina. (2017). Improving BPMN2 Business Process Model to SBVR Business Vocabulary and Business Rules Transformation with BPMN2 Event Naming Patterns. *Information Technology And Control*. 45. 10.5755/j01.itc.45.4.14965.

95 Linehan, Mark. (2007). Ontologies and Rules in Business Models. 149-156. 10.1109/EDOCW.2007.23.

96 De Tommasi, Maurizio & Corallo, Angelo. (2006). SBEAVER: A tool for modeling business vocabularies and business rules. 4253. 1083-1091. 10.1007/11893011_137.

97 Bajwa, Imran & Lee, Mark & Bordbar, Behzad. (2011). SBVR Business Rules Generation from Natural Language Specification. *AAAI Spring Symposium - Technical Report.*

98 Al Khalil, Firas & Ceci, Marcello & Yapa Bandara, PhD, Kosala & O'Brien, Leona. (2016). SBVR to OWL 2 Mapping in the Domain of Legal Rules. 9718. 258-266. 10.1007/978-3-319-42019-6_17.

99 Linehan, Mark. (2008). SBVR Use Cases.. 182-196. 10.1007/978-3-540-88808-6_20.

100Bajwa, Imran & Bordbar, Behzad & Lee, Mark. (2011). SBVR vs OCL: A comparative analysis of standards. *Proceedings of the 14th IEEE International Multitopic Conference 2011*, INMIC 2011. 10.1109/INMIC.2011.6151485.

101 Chittimalli, Pavan & Bhattacharyya, Abhidip. (2019). SBVR-based Business Rule Creation for Legacy Programs using Variable Provenance. 1-11. 10.1145/3299771.3299786.

102 Bollen, Peter. (2008). SBVR: A Fact-Oriented OMG Standard. 718-727. 10.1007/978-3-540-88875-8_96.

103 Agrawal, Ashish. (2011). Semantics of business process vocabulary and process rules. 61-68. 10.1145/1953355.1953363.

104 Boussadi, Abdelali & Bousquet, Cedric & Sabatier, Brigitte & Colombet, Isabelle & Degoulet, Patrice. (2008). Specification of business rules for the development of hospital alarm system: Application to the pharmaceutical validation. *Studies in health technology and informatics*. 136. 145-50. 10.3233/978-1-58603-864-9-145.

105 de Sainte Marie, Christian & Iglesias Escudero, Miguel & Rosina, Peter. (2011). The ONTORULE Project : Where Ontology Meets Business Rules. 24-29. 10.1007/978-3-642-23580-1_3.

106 Koehler, Jana. (2011). The process-rule continuum - Can BPMN & SBVR cope with the challenge?. 302 - 309. 10.1109/CEC.2011.22.

107 Kluza, Krzysztof & Nalepa, Grzegorz. (2013). Towards rule-oriented business process model generation. 2013 *Federated Conference on Computer Science and Information Systems, FedCSIS* 2013. 939-946.

108 Mickeviciute, Egle & Butleris, Rimantas. (2013). Towards the Combination of BPMN Process Models with SBVR Business Vocabularies and Rules. Communications in Computer and Information Science. 403. 114-121. 10.1007/978-3-642-41947-8_11.

109 Mickeviciute, Egle & Butleris, Rimantas & Gudas, Saulius & Karčiauskas, Eimutis. (2017). Transforming BPMN 2.0 Business Process Model into SBVR Business Vocabulary and Rules. *Information Technology And Control*. 46. 10.5755/j01.itc.46.3.18520.

110 Levy Siqueira, Fabio & C. de Sousa, Thiago & Muniz Silva, Paulo. (2017). Using BDD and SBVR to Refine Business Goals into an Event-B Model: A Research Idea. 31-36. 10.1109/FormaliSE.2017.5.

111 Musham, Prakash & Singh, Samiksha & Bahal, Rashi & tv, Prabhu. (2008). Visual SBVR. 676 - 683. 10.1109/ICDIM.2008.4746768.

112 Guissé, Abdoulaye & Lévy, François & Nazarenko, Adeline. (2012). From Regulatory Texts to BRMS: How to Guide the Acquisition of Business Rules?. 10.1007/978-3-642-32689-9_7.

113 zur Muehlen, Michael & Indulska, Marta. (2009). Modeling Languages for Business Processes and Business Rules: A Representational Analysis. *Information Systems*. 35. 379-390. 10.1016/j.is.2009.02.006.

114 zur Muehlen, Michael & Indulska, Marta & Kamp, Gerrit. (2007). Business Process and Business Rule Modeling: A Representational Analysis. 189 - 196. 10.1109/EDOCW.2007.8.

115 Goedertier, Stijn & Vanthienen, Jan. (2007). Declarative Process Modeling with Business Vocabulary and Business Rules. 603-612. 10.1007/978-3-540-76888-3_83.

116 Boussadi, Abdelali & Bousquet, Cedric & Sabatier, B & Caruba, T & Durieux, Pierre & Degoulet, Patrice. (2010). A Business Rules Design Framework for a Pharmaceutical Validation and Alert System. *Methods of information in medicine*. 50. 36-50. 10.3414/ME09-01-0074.

117 Barmpis, Konstantinos & Kolovos, Dimitrios & Hingorani, Justin. (2018). Towards a framework for writing executable natural language rules.

118 Krisciuniene, Gintare & Nemuraite, Lina & Butkienė, Rita & Paradauskas, Bronius. (2013). Developing SBVR Vocabularies and Business Rules from OWL 2 Ontologies. *Communications in Computer and Information Science*. 403. 10.1007/978-3-642-41947-8_13.

119 Uzuncaova, Engin & Khurshid, Sarfraz. (2008). Constraint Prioritization for Efficient Analysis of Declarative Models. 310-325. 10.1007/978-3-540-68237-0_22.

120 Shlyakhter, Ilya & Seater, Robert & Jackson, Daniel & Sridharan, Manu & Taghdiri, Mana. (2004). Debugging Overconstrained Declarative Models Using Unsatisfiable Cores.

121 Rychkova, Irina & Regev, G & Wegmann, Alain. (2008). High-Level Design and Analysis of Business Processes. The Advantages of Declarative Specifications. 99 - 110. 10.1109/RCIS.2008.4632098.

122 Bragaglia, Stefano & Chesani, Federico & Fry, Emory & Mello, Paola & Montali, Marco & Sottara, Davide. (2011). Event Condition Expectation (ECE-) Rules for Monitoring Observable Systems. 7018. 267-281. 10.1007/978-3-642-24908-2_28.

123 Kliegr, Tomáš & Kuchař, Jaroslav & Sottara, Davide & Vojíř, Stanislav. (2014). Learning Business Rules with Association Rule Classifiers. 10.13140/2.1.1737.9208.

124 Bragaglia, Stefano & Chesani, Federico & Ciampolini, Anna & Mello, Paola & Montali, Marco & Sottara, Davide. (2010). An Hybrid Architecture Integrating Forward Rules with Fuzzy Ontological Reasoning. 6076. 438-445. 10.1007/978-3-642-13769-3_53.

125 Ezekiel, Kanana & Vassilev, Vassil & Ouazzane, Karim & Patel, Yogesh. (2018). Adaptive business rules framework for workflow management. Business Process Management Journal. 10.1108/BPMJ-08-2017-0219.

126 Dang, Jiangbo & Hedayati, Amir & Hampel, Ken & Toklu, Candemir. (2008). An ontological knowledge framework for adaptive medical workflow. Journal of biomedical informatics. 41. 829-36. 10.1016/j.jbi.2008.05.012.

127 Pham, Tuan Anh & Thanh, Nhan. (2015). Checking the Compliance of Business Processes and Business Rules Using OWL 2 Ontology and SWRL. 427. 10.1007/978-3-319-29504-6_3.

128 Costello, Claire & Molloy, Owen. (2008). Towards a Semantic Framework for Business Activity Monitoring and Management.. 17-27.

129 Rosenberg, Florian & Nagl, Christoph & Dustdar, Schahram. (2006). Applying distributed business rules - The VIDRE approach. 471-478. 10.1109/SCC.2006.22.

130Nagl, Christoph & Rosenberg, Florian & Dustdar, Schahram. (2006). VIDRE - A distributed service-oriented business rule engine based on RuleML. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*. 35-44. 10.1109/EDOC.2006.67.

131 Palmirani, Monica & Governatori, Guido & Rotolo, Antonino & Tabet, Said & Boley, Harold & Paschke, Adrian. (1970). LegalRuleML: XML-Based Rules and Norms. 10.1007/978-3-642-24908-2_30.

132 Arévalo, Carlos & Escalona, M.J. & Ramos, Isabel. (2016). A metamodel to integrate business processes time perspective in BPMN 2.0. *Information and Software Technology*. 77. 10.1016/j.infsof.2016.05.004.

133 Kluza, Krzysztof & Nalepa, Grzegorz. (2017). A Method for Generation and Design of Business Processes with Business Rules. Information and Software Technology. 10.1016/j.infsof.2017.07.001.

134 Mens, Joris & Luiten, Sander & Driel, Yannick & Smit, Jakobus & Ravesteyn, Pascal. (2015). Standardisation of risk screening processes in healthcare through business rules management.

135 Longo, A., Bochicchio, M. A., Zappatore, M., & Vaira, L. (2018). Discussion Paper: Filling the Gap Between Business Rules and Technical Requirements in Business Analytics: The Fact - Centered ETL Approach. SEBD.

136 Skersys, Tomas & Butleris, Rimantas & Kapocius, Kestutis & Vileiniskis, Tomas. (2013). An Approach for Extracting Business Vocabularies from Business Process Models. *Information technology and control*. 42. 178-190. 10.5755/j01.itc.42.2.2310.

137 De Giacomo, Giuseppe & Dumas, Marlon & Maggi, Fabrizio & Montali, Marco. (2015). Declarative Process Modeling in BPMN. 84-100. 10.1007/978-3-319-19069-3_6.

138Döhring, Markus & Zimmermann, Birgit & Karg, Lars. (2011). Flexible Workflows at Design- and Runtime Using BPMN2 Adaptation Patterns. 87. 25-36. 10.1007/978-3-642-21863-7_3.

139 Milanović, Milan & Gasevic, Dragan & Rocha, Luis. (2011). Modeling Flexible Business Processes with Business Rule Patterns. *Proceedings - IEEE*

International Enterprise Distributed Object Computing Workshop, EDOC. 65 - 74. 10.1109/EDOC.2011.25.

140 Milanović, Milan & Gasevic, Dragan. (2009). Towards a Language for Rule-Enhanced Business Process Modeling. *Proceedings - 13th IEEE International Enterprise Distributed Object Computing Conference, EDOC* 2009. 64-73. 10.1109/EDOC.2009.12.

141 Döhring, Markus & Zimmermann, Birgit. (2011). vBPMN: Event-aware workflow variants by weaving BPMN2 and business rules. *Lecture Notes in Business Information Processing*. 81. 332-341. 10.1007/978-3-642-21759-3_24.

142 De Giacomo, Giuseppe & Marrella, Andrea & Patrizi, Fabio & Maggi, Fabrizio. (2017). On the Disruptive Effectiveness of Automated Planning for LTLf-based Trace Alignment.

143 Lam, Vitus. (2017). Detecting violation of business constraints in declarative process execution: a case study. *Business Process Management Journal.* 23. 00-00. 10.1108/BPMJ-05-2016-0105.

144 Räim, Margus & Di Ciccio, Claudio & Maggi, Fabrizio & Mecella, Massimo & Mendling, Jan. (2014). Log-Based Understanding of Business Processes through Temporal Logic Query Checking. 10.13140/2.1.2100.6080.

145 De Giacomo, Giuseppe & De Masellis, Riccardo & Grasso, Marco & Maggi, Fabrizio & Montali, Marco. (2014). Monitoring Business Metaconstraints Based on LTL and LDL for Finite Traces. 1-17. 10.1007/978-3-319-10172-9_1.

146 Knolmayer, G., Endl, R., & Pfahrer, M. (2000). Modeling Processes and Workflows by Business Rules. *Business Process Management*.

147 Boukhebouze, Mohamed, et al. “A Rule-Based Approach to Model and Verify Flexible Business Processes.” *International Journal of Business Process Integration and Management*, vol. 5, no. 4, 2011, p. 287, 10.1504/ijbpim.2011.043389.

148 Giordano, Laura & Martelli, Alberto & Spiotta, Matteo & Dupré, Daniele. (2013). Business processes verification with temporal answer set programming. *Theory and Practice of Logic Programming*. 13. 10.1017/S1471068413000409.

149 Papazoglou, Mike. (2011). Making Business Processes Compliant to Standards & Regulations. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*. 3 - 13. 10.1109/EDOC.2011.37.

150 Deesukying, Jatuporn & Vatanawood, Wiwat. (2016). Generating of business rules for Coloured Petri Nets. 1-6. 10.1109/ICIS.2016.7550824.

151 Ghanbari Ghooshchi, Nina & Beest, Nick & Governatori, Guido & Olivieri, Francesco & Sattar, Abdul. (2017). Visualisation of Compliant Declarative Business Processes. 10.1109/EDOC.2017.21.

152 Baier, Thomas & Di Ciccio, Claudio & Mendling, Jan & Weske, Mathias. (2015). Matching of Events and Activities - An Approach Using Declarative Modeling Constraints. *Lecture Notes in Business Information Processing*. 214. 10.1007/978-3-319-19237-6_8.

153 Olivieri, Francesco & Governatori, Guido & Beest, Nick & Ghanbari Ghooshchi, Nina. (2018). Declarative Approaches for Compliance by Design. 10.1007/978-3-319-76587-7_6.

154 Song, R., Wang, Y., Cui, W., Vanthienen, J., & Huang, L. (2018). Towards Improving Context Interpretation in the IoT Paradigm: a Solution to Integrate Context Information in Process Models. ICMSS 2018.

155 Goedertier, Stijn & Vanthienen, Jan. (2007). A Vocabulary and Execution Model for Declarative Service Orchestration. 496-501. 10.1007/978-3-540-78238-4_50.

156 Nalepa, Grzegorz. (2007). Proposal of Business Process and Rules Modeling with the XTT Method. 500 - 506. 10.1109/SYNASC.2007.58.

157 Kluza, Krzysztof & Kaczor, Krzysztof & Nalepa, Grzegorz. (2012). Enriching Business Processes with Rules using the Oryx BPMN Editor. 573-581. 10.1007/978-3-642-29350-4_68.

158 Kluza, Krzysztof & Maslanka, Tomasz & Nalepa, Grzegorz & Ligęza, Antoni. (2011). Proposal of Representing BPMN Diagrams with XTT2-Based Business Rules. 382. 243-248. 10.1007/978-3-642-24013-3_25.

159 Kluza, Krzysztof & Nalepa, Grzegorz. (2018). Formal Model of Business Processes Integrated with Business Rules. *Information Systems Frontiers*. 10.1007/s10796-018-9826-y.

160 Nalepa, Grzegorz & Kluza, Krzysztof & Ernst, Sebastian. (2011). Modeling and Analysis of Business Processes with Business Rules.

161 Wang, Olivier & Ke, Changhai & Liberti, Leo & de Sainte Marie, Christian. (2016). The Learnability of Business Rules. 10122. 257-268. 10.1007/978-3-319-51469-7_22.

162 Rachdi, Anass & En-Nouaary, Abdeslam & Dahchour, Mohamed. (2016). Verification of Common Business Rules in BPMN Process Models. 9944. 334-339. 10.1007/978-3-319-46140-3_27.

163 Rachdi, A., En-Nouaary, A., & Dahchour, M. (2016). Analysis of common business rules in BPMN process models using business rule language. 2016 *11th International Conference on Intelligent Systems: Theories and Applications* (SITA), 1-6.

164 Vaculín, Roman & Hull, Richard & Heath, Fenno & Cochran, Craig & Nigam, Anil & Sukaviriya, Noi. (2011). Declarative business artifact centric modeling of decision and knowledge intensive business processes. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*. 151 - 160. 10.1109/EDOC.2011.36.

165 Eshuis, Rik. (2018). Feature-Oriented Composition of Declarative Artifact-Centric Process Models: 16th International Conference, BPM 2018, Sydney, NSW, Australia, September 9–14, 2018, Proceedings. 10.1007/978-3-319-98648-7_5.

166 Eshuis, Rik & Hull, Richard & Sun, Yutian & Vaculín, Roman. (2014). Splitting GSM Schemas: A Framework for Outsourcing of Declarative Artifact Systems. Information Systems. 46. 157–187. 10.1016/j.is.2014.04.005.

167 Valatkaite, Irma & Vasilecas, Olegas. (2005). On Business Rules Automation: The BR-Centric IS Development Framework. 349-364. 10.1007/11547686_26.

168 Valatkaite, Irma & Vasilecas, Olegas. (2003). A Conceptual Graphs Approach for Business Rules Modeling. 2798. 178-189. 10.1007/978-3-540-39403-7_15.

169 Governatori, Guido & Rotolo, Antonino. (2008). An Algorithm for Business Process Compliance. Knowledge and Information Systems - KAIS. 186-191. 10.3233/978-1-58603-952-3-186.

170 Kamada, Aqueo & Governatori, Guido & Sadiq, Shazia. (2010). Transformation of SBVR Compliant Business Rules to Executable FCL Rules. 6403. 153-161. 10.1007/978-3-642-16289-3_14.

171 Becker, J., Ahrendt, C., Coners, A., Weiß, B., & Winkelmann, A. (2010). Business Rule Based Extension of a Semantic Process Modeling Language for Managing Business Process Compliance in the Financial Sector. GI Jahrestagung.

172 Becker, Jörg & Ahrendt, Christoph & Coners, André & Weiß, Burkhard & Winkelmann, Axel. (2011). Modeling and Analysis of Business Process Compliance. Governance and Sustainability in Information Systems. 366. 259-269. 10.1007/978-3-642-24148-2_17.

173 Ali, S., Soh, B., & Torabi, T. (2006). A novel approach toward integration of rules into business processes using an agent-oriented framework. *IEEE Transactions on Industrial Informatics*, 2, 145-154.

174 Taveter, Kuldar & Wagner, Gerd. (2001). Agent-Oriented Enterprise Modeling Based on Business Rules. 527-540. 10.1007/3-540-45581-7_39.

175 Lazovik, Alexander & Aiello, Marco & Papazoglou, Mike. (2004). Associating assertions with business processes and monitoring their execution. ICSOC '04: *Proceedings of the Second International Conference on Service Oriented Computing*. 94-104. 10.1145/1035167.1035182.

176 De Nicola, Antonio & Missikoff, Michele & Smith, Fabrizio. (2012). Towards a method for business process and informal business rules compliance. *Journal of Software: Evolution and Process*. 24. 341-360. 10.1002/smr.553.

177 Sun, Yuqing & Zhexue Huang, Joshua & Meng, Xiangxu. (2011). Integrating constraints to support legally flexible business processes. *Information Systems Frontiers*. 13. 171-189. 10.1007/s10796-009-9190-z.

178 Mulyar, Nataliya & Pesic, Maja & Aalst, Wil M. P. & Peleg, Mor. (2007). Declarative and Procedural Approaches for Modelling Clinical Guidelines: Addressing Flexibility Issues. Allergy. 4928. 335-346. 10.1007/978-3-540-78238-4_35.

179 Rychkova, Irina & Nurcan, Selmin. (2011). Towards Adaptability and Control for Knowledge-Intensive Business Processes: Declarative Configurable Process Specifications. 1 - 10. 10.1109/HICSS.2011.452.

180 Zugal, Stefan & Pinggera, Jakob & Weber, Barbara. (2011). The Impact of Testcases on the Maintainability of Declarative Process Models. *Lecture Notes in Business Information Processing*. 81. 163-177. 10.1007/978-3-642-21759-3_12.

181 Bellodi, Elena & Riguzzi, Fabrizio & Lamma, Evelina. (2010). Probabilistic Declarative Process Mining. Proc. 4th Int. Conf. Knowl. Sci. Eng. Manage.. 292-303. 10.1007/978-3-642-15280-1_28.

182 Bellodi, Elena & Riguzzi, Fabrizio & Lamma, Evelina. (2010). Probabilistic Logic-based Process Mining. CEUR Workshop Proceedings. 598.

183 Wondoh, John & Grossmann, Georg & Stumptner, Markus. (2017). Dynamic temporal constraints in business processes. 1-10. 10.1145/3014812.3014848.

184 Igler, Michael & Moura, Paulo & Zeising, Michael & Jablonski, Stefan. (2010). ESProNa: Constraint-Based Declarative Business Process Modeling. 91 - 98. 10.1109/EDOCW.2010.10.

185 Al-Ali, Hamda & Damiani, Ernesto & Al-Qutayri, Mahmoud & Abu Matar, Mohammad & Mizouni, Rabeb. (2018). Translating BPMN to Business Rules. 10.1007/978-3-319-74161-1_2.

186 Guns, Tias & Dries, Anton & Nijssen, Siegfried & Tack, Guido & De Raedt, Luc. (2015). MiningZinc: A declarative framework for constraint-based mining. Artificial Intelligence. 10.1016/j.artint.2015.09.007.

187 Mrasek, Richard & Mülle, Jutta & Böhm, Klemens. (2015). Automatic Generation of Optimized Process Models from Declarative Specifications. 382-397. 10.1007/978-3-319-19069-3_24.

188 Goedertier, Stijn & Vanthienen, Jan & Regev, G & Soffer, Pnina & Schmidt, R. (2006). Compliant and flexible business processes with business rules. *Katholieke Universiteit Leuven*, Open Access publications from Katholieke Universiteit Leuven. 236.

189 Kumar, Akhil & Yao, Wen. (2009). Process Materialization Using Templates and Rules to Design Flexible Process Models. 122-136. 10.1007/978-3-642-04985-9_13.

190 Groefsema, Heerko & Bulanov, Pavel & Aiello, Marco. (2011). Declarative Enhancement Framework for Business Processes. 7084. 495-504. 10.1007/978-3-642-25535-9_34.

191 Milanović, Milan & Kaviani, Nima & Gasevic, Dragan & Giurca, Adrian & Wagner, Gerd & Devedzic, Vladan & Hatala, Marek. (2007). Business Process Integration by Using General Rule Markup Language. 353-353. 10.1109/EDOC.2007.18.

192 Bragaglia, Stefano & Chesani, Federico & Mello, Paola & Montali, Marco & Torroni, Paolo. (2012). Reactive Event Calculus for Monitoring Global Computing Applications. 7360. 123-146. 10.1007/978-3-642-29414-3_8.

193 Wedemeijer, Lex. (2016). A Metamodel for Business Rules with Access Control. 10.5220/0006222100460053.

194 Lévy, François & Guissé, Abdoulaye & Nazarenko, Adeline & Omrane, Nouha & Szulman, Sylvie. (2010). An Environment for the Joint Management of Written Policies and Business Rules. Proceedings ICTAI. 2. 142 - 149. 10.1109/ICTAI.2010.95.

195 Brzezinski, Jerzy & Danilecki, Arkadiusz & Flotyński, Jakub & Kobusinska, Anna & Stroiński, Andrzej. (2012). ROsWeL Workflow Language: A Declarative, Resource-oriented Approach. *New Generation Computing*. 30. 10.1007/s00354-012-0203-y.

196 Fortineau, Virginie & Fiorentini, Xenia & Paviot, Thomas & Louis Sidney, Ludovic & Lamouri, Samir. (2014). Expressing formal rules within ontology-based models using SWRL: An application to the nuclear industry. *International Journal of Product Lifecycle Management*. 7. 75. 10.1504/IJPLM.2014.065458.

197 Padilha, Bruno & Liberato, Rafael & Schwerz, André & Pu, Calton & Ferreira, João. (2018). WED-SQL: An Intermediate Declarative Language for PAIS Execution. 10.1007/978-3-319-94289-6_26.

198 Meng, Jie & Su, S.Y.W. & Lam, H & Helal, A. (2002). Achieving Dynamic Inter-organizational Workflow Management by Integrating Business Processes. 10 pp.-. 10.1109/HICSS.2002.993858.

199 Hummer, Waldemar & Gaubatz, Patrick & Strembeck, Mark & Zdun, Uwe & Dustdar, Schahram. (2013). Enforcement of Entailment Constraints in Distributed Service-Based Business Processes. *Information and Software Technology*. 55. 10.1016/j.infsof.2013.05.001.

200 Kobti, Ziad & Sundaravadanam, Menaka. (2006). An enhanced conceptual framework to better handle business rules in process oriented applications. ICWE'06: *The Sixth International Conference on Web Engineering*. 273-280. 10.1145/1145581.1145636.

201 Pourmirza, Shaya, et al. “A Systematic Literature Review on the Architecture of Business Process Management Systems.” *Information Systems*, vol. 66, June 2017, pp. 43–58, 10.1016/j.is.2017.01.007.

202 Bernotaityte, Gintare, et al. “Developing SBVR Vocabularies and Business Rules from OWL2 Ontologies.” *Communications in Computer and Information Science*, 2013, pp. 134–145, link.springer.com/chapter/10.1007/978-3-642-41947-8_13, 10.1007/978-3-642-41947-8_13. Accessed 14 Dec. 2019.

203 Szelągowski, Marek. (2018). Evolution of the BPM Lifecycle. 205-211. 10.15439/2018F46.

204 Ampatzoglou, A., Bibi, S., Avgeriou, P., Verbeek, M., & Chatzigeorgiou, A. (2019). Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Inf. Softw. Technol., 106*, 201-230.

8. License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Fodio Tureta Abubakar**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

A Systematic Literature Review on Business Rule Modeling,

supervised by **Fabrizio Maria, Maggi Fredrik Payman Milani and Marcelo Sarini.**

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Fodio Tureta Abubakar
18/05/2020