A Functional Architecture for the Elicitation, Design and Specification of Business Decisions and Business Logic

Koen Smit
HU University of Applied Sciences Utrecht
Utrecht, the Netherlands
e-mail: koen.smit@hu.nl

Sam Leewis
HU University of Applied Sciences Utrecht
Utrecht, the Netherlands
e-mail: sam.leewis@hu.nl

Martijn Zoet
Zuyd University of Applied Sciences
Sittard, the Netherlands
e-mail: martijn.zoet@zuyd.nl

***Abstract*—Functional architectures are created to be used as a standard by their respective industry. Organizations use reference functional architectures to guide their development or as a means to become compliant. However, a reference functional architecture to manage business decisions and business logic does not yet exist. One research field that focuses on the management of business decisions and business logic is Business Rules Management (BRM). By analyzing the functional requirements of seven Dutch governmental institutions with regards to the elicitation, design and specification of business decisions and business logic, we aim to propose the first version of a reference functional architecture for BRM. To do so, we utilized three thematic coding rounds to analyze 536 functional requirements for BRM solutions, resulting in 18 functional categories and mapped the functional categories to the BRM capabilities. The results form a first basis for the construction of a reference functional architecture for BRM capabilities, also identifying multiple directions for future research.**

Keywords-Business Rules Management; Functional Architecture; Functional Requirements.

I. INTRODUCTION

Decisions are amongst the most important assets of an organization [1], and business decisions and business logic are an important part of an organization's daily activities. Therefore, the performance of an organization depends on the ability to manage its business decisions and business logic [2].

To structure the process of managing business decisions and business logic, Business Rules Management (BRM) could be utilized. BRM comprises a systematic and controlled approach to support the elicitation, design, specification, verification, validation, deployment, execution, governance, and monitoring of business decisions and business logic [3]–[6], see Figure 1.

Considering the BRM research domain, a predominant focus towards technically-oriented research can be identified. For example, Nelson et al. [7] state: "*studies provide beginnings of a business rules research program, but collectively the research often overlooks major steps in BRM and fails to focus on business rules specific challenges and the larger context that rules play in organizations.*" Moreover, Kovacic [8] argues about the current research directions in the BRM research field, stating: "*With so much emphasis towards the technological aspects, we can lose sight of the management of information systems considerations.*"

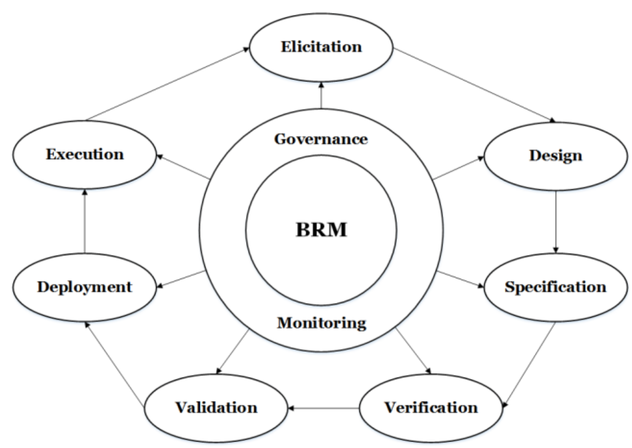

Figure 1. BRM capability overview [3]–[6].

Therefore, we identify that there is an imbalance when comparing technical-oriented research to the management of information systems and BRM artefacts used in BRM processes. In addition, in the work of Arnott and Pervan [9] featuring a thorough literature review, a conclusion is drawn stating that the field has lost its connection with industry some time ago and research input with practical relevance is scarce. Arnott and Pervan revisited the knowledge base in 2014 and concluded that a transition is taking place towards a more practical-oriented approach, whilst a strong connection between theory and practice is still lacking [10], which is also concluded in the work of Zoet [11]. Therefore, research conducted in the area of BRM should also ground practical usability, while taking into account the theory as part of the existing academic knowledge base on BRM.

In this study, we revisit an extended version of our dataset containing functional requirements from seven large Dutch governmental institutions intending to derive a functional architecture that other organizations could utilize to design BRM solutions. In an earlier study [12] we identified that some research has been conducted on BRM-related functional architectures. For example, Schlosser, Baghi, Otto and Oesterle [5] propose three architectural perspectives that could guide organizations designing BRM solutions, however, do so at a high-level of abstraction. Our previous study, in which we analyzed a set of 750 BRM-related functional requirements, resulted in a functional framework in which we identified several themes per BRM capability [12]. This study seeks to extend the understanding of functional requirements, in the context of BRM, by exploring the required functionalities for Business Rules Management Systems. This paper focuses on the first three BRM capabilities, being elicitation, design, and specification. To do so, we aim to answer the following research question: “Which functional requirement categories should be taken into account when designing a BRM functional architecture for the elicitation, design and specification capabilities?”

The remainder of this paper is structured as follows: In Section 2, we provide insights into the elicitation, design and specification BRM capabilities, as well as the value of functional architectures in the domain of BRM. In Section 3, the research method that was utilized to collect and analyze the data required to construct the functional architectures is described. In Section 4, the manner in which the data is collected, as well as analyzed is presented. In Section 5, the functional architectures are presented and elaborated in the results Section. In Section 6, we discuss the conclusions of our research and provide discussion about our research method and results. Section 7 presents possible directions for future results.

II. BACKGROUND AND RELATED WORK

Organizations are increasingly looking for ways to automate products and services. Doing so, organizations need to ensure that these products and services take into account all legal sources that influence the organization doing business, i.e., law, regulations, internal policies or international conventions [13]. To do so, business decisions and underlying business logic are implemented. Business decisions and business logic are an important part of an organization’s daily activities. A business decision is defined as: “A conclusion that a business arrives at through business logic and which the business is interested in managing” [14]. Moreover, business logic is defined as: “a collection of business rules, business decision tables, or executable analytic models to make individual business decisions” [15]. In theory and practice, business decisions and business logic comprise several different concepts, such as derivation structures, decision tables, business vocabularies, fact type models and rule requirements [16], [17]. However, as our focus in this paper is not to define these different concepts that are utilized in a variety of ways by organizations, we adhere to these concepts as artifacts in a general sense. Example artefacts (i.e., sources, contexts and business rules) used to define and implement business decisions and business logic are depicted in Figure 2. See, for a detailed description of each of the concepts to design, specify, and execute business decisions and business logic in the work of Smit and Zoet [17]. When individual artefacts are affected in the functional category, the artefact is specified with a label, e.g., ‘*derivation structure*’. However, when it concerns the collection of all artefacts, the general term ‘artefact’ is used in this paper.

The previous section already mentioned the specific focus of this study on the elicitation, design and specification BRM capabilities. Based on the definition of [15], a capability is defined as: “*An ability that an organization, person, or system, possesses.*” A detailed explanation of each capability can be found in [6], [17]. However, to ground our research, a summary of the elicitation, design, specification, verification, deployment, execution, governance and monitoring capabilities is provided here.

The purpose of the elicitation capability is twofold. First, the purpose is to determine the knowledge that needs to be captured from various legal sources to realize the value proposition of the business rules [18]. Different types of legal sources from which knowledge can be derived are, for example, laws, regulations, policies, internal documentation, guidance documents, parliament documents, official disclosures, implementation instructions, and experts. Depending on the type of knowledge source(s), for example, documentation versus experts, different methods, processes, techniques and tools to extract the knowledge are applied [19]. The second purpose is to conduct an impact analysis is if a business rule architecture is already in place. When all relevant knowledge is captured, the business decisions need to be designed in the design capability. The purpose of the design capability is to establish a business rules architecture, which contains the business decisions and how the business decisions are derived to deliver the value proposition [16]. After the business rule architecture is designed, the contents of the business decisions need to be specified in the specification capability. The purpose of the specification capability is to write the business logic and create the fact types needed to define or constrain some particular aspect of the business. After the business logic is created, it is verified and validated in the verification and validation capabilities, respectively.

The capabilities described are implemented by organizations in different ways. One common approach is to implement information systems that are tailored to one or a combination of the elicitation, design and specification capabilities. Such information systems are often referred to as Business Rules Management Systems (BRMS) [18][19]. Looking at the architecture of Information systems, decomposition can be achieved by the creation of several different architectural views or perspectives, i.e., technical, functional, information, data, process, components, service or classes [20]. Analysis of the BRM body of knowledge shows that the functional perspective, also referred to as the functional architecture, has received little attention [5][7], [8] compared to the technical perspectives. The functional architecture perspective is equally important compared to the other perspectives as it guides, especially business stakeholders, with the exact functionality an information system offers to execute a capability. Developing a functional architecture for BRM capabilities is therefore also in line with the lack of research in the BRM domain that is practically oriented [9]–[11]. In this paper we adhere to the following definition of a functional architecture: a functional architecture comprises a modular decomposition of the functionality of an information system [21].

Functional Architecture of software products, which comprises: a modular decomposition of the product functionality; a simple notation for easy comprehension by non-specialists; and applicability in any line of business, offering a uniform method for modeling the functionalities of software products [21]. Functional architecture perspectives are, for example, utilized in practice by integrating them in standard operating models [22][23]. Examples of such models are the eTOM business process framework [24], the Insurance Application Architecture (IAA) [25] and the Banking Industry Architecture Network (BIAN) [26]. Functional architectures for BRM can be established using both inductive, as well as deductive reasoning. The current body of knowledge does not contain detailed contributions to help the construction of a functional architecture for the BRM capabilities. Therefore, the approach in this paper follows an inductive approach to construct a BRM functional architecture from the BRM-related requirements that are collected. In this paper, we solely focus on functional requirements with regards to BRM systems as a functional requirement emphasizes what is required, and not how. This is in line with the notion of a capability, which also focuses on what (value) an organization can deliver, but not how the value is delivered. The functional requirements are often created by subject-matter experts, which are also the stakeholders and end-users of the BRMS that is being designed or developed. This strengthens the validity of the resulting functional architecture.

In literature and practice, several methods exist to formulate functional requirements, i.e., personas, wireframing, use cases, mockups, and user stories [27]. User stories are increasingly being adopted and are comprehensible by, i.e., both developers and customers and support participatory design by all stakeholders as they are all able to design the behavior of the system. In addition to user stories, the agile community [28] also utilize epics and themes. An epic is a large user story while a theme is a collection of user stories. Making use of user stories enables empirical-focused design by enabling the designers to make decisions by studying prospective users in typical situations [28]. The organizations analyzed all defined their functional requirements employing user stories. Therefore, in our study, the unit of analysis is a user story.

III. RESEARCH METHOD

This research aims at creating a functional architecture containing the BRM capabilities: Elicitation, Design, and Specification. Therefore, qualitative research is selected as our research methodology. Case study research is chosen as the most suitable strategy for this research.

By selecting case study research, the researchers were able to gather functional requirements for the BRM capabilities Elicitation, Design, and Specification in the Dutch public sector. Our study utilizes a holistic case study approach, more on this in the work of Yin [29]. This case study approach features one context, BRM solutions requirements phase, and four cases containing in this context. The BRM solution-related set of functional requirements of the participating organizations is set as the unit of analysis. The data collection consisted of secondary data, which is a form of third-degree data collection. According to [30], when data, such as requirements are studied, third-degree data collection is the best fit. The coding of the functional architecture consists of three rounds of coding according to Strauss and Corbin's process of open coding, axial coding, and selective coding [31].

IV. DATA COLLECTION AND ANALYSIS

The data collection for this study occurred for eleven months, between November 2016 and September 2017. The selection of the participants is based on the group of individuals, organizations, information technology, or community which best represents the studied phenomenon [31]. Related to this study, the studied phenomenon is represented by organizations, and individuals within these organizations, which deal with the selection of BRM solution-related requirements. Organizations dealing with these BRM solution-related requirements are often financial and government institutions because of the large-transaction, knowledge-intensive, digital products and services they deliver. Therefore, several Dutch executive governmental agencies were invited to provide requirements for this study. Executive governmental agencies are responsible for the execution of a variety of services like the screening of immigrants, handling student loans, tax returns etc. thereby serving approximately 17 million citizens and organizations in the Netherlands. The participating governmental agencies are comparable in terms of business processes. The participating seven governmental agencies requested that their data is handled anonymously. Therefore, from this moment on, the organizations are labelled as A, B, C, D, E, F, and G, as shown in Table I. The participating organizations were invited to gather and send all their BRM solutions-related requirement documentation to the researchers. Each organization defined their BRM solution-related requirements with a team existing of an enterprise architect, business rules architect, business rules analyst, legal or policy expert. Additional support was provided by a procurement officer, BRM project manager, business consultant, IT architect and external advisors.

Based on the data received, the researchers analysed and structured the functional requirements. The data analysis consisted of three rounds of thematic coding, according to Strauss and Corbin's process of 1) open coding, 2) axial coding, and 3) selective coding [31]. During the coding rounds, two researchers coded separately from each other thereby increasing the inter-reliability in the coding [32].

The first round of coding is the open coding round. The open coding round identifies the functional requirements from the secondary data together with the meta-data of the functional requirements. To ensure optimal analysis the researchers numbered each requirement with a unique ID. Additionally, for each requirement the responsible role (i.e., manager or business rule analyst) was added, the feature (what does the owner or role wants with the functionality), the feature outcome (the benefit of the functionality), organization and an organization ID (to ensure the traceability of the functional requirement towards the case organization documents). During this round of coding, two situations occurred: 1) The functional requirements could be documented explicitly by registering the organization name and organization ID, as shown in Table I, or II) the functional requirements were stated implicitly as nested requirements or plain text.

The second round of coding is the axial coding round. Axial coding refines and differentiates concepts that are already available and code them into categories [33]. The axial coding round was utilized to structure the functional requirements over the BRM capabilities Elicitation, Design and Specification proposed by [6][17]. Therefore, the coding scheme in this round is as follows: Elicitation, Design, and Specification. For example, the two requirements in Table I are coded into the Elicitation capability.

The third and thereby last round of coding is selective coding. The purpose of the selective coding round is the identification of functional categories [33]. This round of coding is focused on the identification of categories within the set of functional requirements distributed over the BRM capabilities in the axial coding round. Our earlier work on functional requirement themes for BRM capabilities is also taken into account in this coding round, which resulted in eleven functional themes [12]. These were (Elicitation) 1) Import Sources, 2) Annotate Sources, 3) Generate Overviews, 4) Perform Impact-Analysis, (Design) 5) Create Business Decisions, 6) Create Relationships, 7) Create Overviews, 8) Reuse Business Decisions, (Specification) 9) Define Business Logic, 10) Add Meta-Data, and 11) Create Relationships. These themes could influence the functional architecture that is being constructed in this paper.

TABLE I. EXAMPLES OF CODED FUNCTIONAL REQUIREMENTS

ID	Role	Feature	Outcome	Organization	Organization reference	Category
7	Rule Analyst	I want an overview of all relevant sources	So that I can scope the project	A	5.3	Create Overview
13	Rule Analyst	I want to be able to include a source in the analysis environ-ment	So that the source is in the system ready for analysis	B	PR13_UR_A_1	Import Sources

Additionally, the coded categories in the three capabilities are checked for possible overlap. An example of this is the category ''impact analysis'' which exists in the Elicitation, Design, and Specification capability

V. RESULTS

In this section, the results of our data collection and analysis are presented and elaborated. Per coding round, as described in the previous section, descriptive results are provided. This is followed by the presentation of the functional architecture and the elaboration of the functional categories it comprises.

TABLE II. BREAKDOWN OF FUNCTIONAL REQUIREMENTS RECEIVED FROM THE CASE ORGANIZATIONS

Organization	Total number of functional requirements identified
A	130
B	52
C	126
D	67
E	123
F	38
G	0

For the construction of the functional architecture, to the knowledge of the authors, no explicit practices or specific guidelines exist. However, to theoretically ground the construction of the functional architecture, several definitions are analyzed that comprise one or multiple characteristics that compose a functional architecture. This leads us to the following criteria [21][34][35]: 1) a functional architecture represents a high-level view of the major functions from a usage perspective, 2) a functional architecture specifies the interactions of functions internally between each other and externally with other products, 3) the functionalities presented represent arrangements of requirements, 4) the functional architecture should be expressed in easy to understand diagrams, 5) the functional architecture should be constructed with the input of relevant stakeholders, such as product managers, architects, and managers.

The open coding resulted in the registration of 536 functional requirements, originating from seven organizations, see Table II.

TABLE III. BREAKDOWN OF FUNCTIONAL REQUIREMENTS PER BRM CAPABILITY

Organization/ Capability	Elicitation		Design		Specification		Total
A	0	0.0%	8	1.6%	122	22.7%	130
B	1	0.1%	4	0.8%	47	8.7%	52
C	12	2.2%	52	9.7%	62	11.5%	126
D	20	3.7%	25	4.6%	22	4.2%	67
E	42	7.8%	14	2.7%	67	12.5%	123
F	1	0.1%	7	1.4%	30	5.7%	38
G	0	0%	0	0%	0	0%	0

Subsequently, the second round of coding consisted of assigning the functional requirements to either the elicitation, design or specification BRM capability as described in the previous section. The results of this process are presented in Table III. In the second coding round, no differences were identified between both researchers.

The third round of coding resulted in the identification of 18 functional requirement categories, see Figure 3. For each functional requirement category, we report on its number, functionality and possible overlap with functionality categories as part of the other BRM capabilities. In the third coding round, 14 differences were identified in the coding and were resolved by the third researcher. Lastly, to better understand the artefacts described in this section, we refer to Section 2 in this paper, as well as the work of Smit and Zoet [6].

A. *Elicitation*

With regards to the elicitation capability, four functional categories were identified: 1) Import Sources, 2) Annotate Sources, 3) Perform Impact Analysis, and 4) Compare Sources.

1) Import Sources - The knowledge needed to create business decisions and business logic is elicitated from a variety of different sources, i.e., laws, regulations, policies, internal documentation, guidance documents, parliament documents, implementation instructions, and official disclosures [12]. This functionality encompasses the import of a source, which must be supported in both manual and automated style. As these sources come in different formats or type of documents, the functionality should support as many as possible extensions that can be imported, i.e., MS Office document types, PDF, XML, other open-source word processors, or HTML. Also, in some source types, tables and figures or other representations are important to take into account. Therefore, functional support for importing media as part of sources is deemed important.

2) Annotate Sources – Concerns the manual annotation of sources used to create business decisions and business logic, i.e., derivation structures, terms, or roles. As organizations all differ significantly from each other in terms of what concepts to annotate in sources, i.e., fact types, sentences or sections, functional support to ensure organizations can modify the concepts to annotate should be taken into account. This also includes the support for definition and use of templates for analysts to use during the annotation process.

3) Compare Sources – Encompasses the functional support to compare two or more sources. This is required by analysts that are tasked to review the changes to legal sources that affect the already implemented business decisions and business logic. Comparison of sources must be supported in an automated way in which the machine recognizes and labels Create, Update, and Delete modification types. Similar to the import source functionality, functional support for multiple document types is essential as these documents need to be compared exactly as published by their source. Functional support for automatic comparison of sources enables the reduction of human error and could boost efficiency because of the decrease in manual comparison.

4) Perform Impact Analysis – Allows the user to determine the impact of modified sources with regards to already implemented business decisions and business logic. This functionality should enable the selection of artefacts to review its dependencies with other artefacts, which, on the one hand, encompasses the support for manual impact analysis. On the other hand, functional support for an automatic impact assessment that enables a user to input scenario variables to calculate the impact should be present as well. Automatic impact assessment is regarded as it allows for higher efficiency and less human error. The results of an impact analysis are often used for communication with stakeholders and to determine a course of action. Therefore, there must be functional support for exporting (part of) the impact assessment results in the format and with the variables that the organization requires.

B. *Design*

With regards to the design capability, two functional categories were identified: 5) Navigate Artefact Structure and 6) Define templates.

5) Navigate Artefact Structure – The roles responsible for creating or modifying business decisions and business logic need to be able to search and navigate efficiently and effectively to be able to do so. This could be achieved in several ways, depending on the requirements of the

 ISBN: 978-1-61208-765-8

organizations, however, the navigation should support the selection of all possible artefacts to view during navigation through business decisions and business logic. While doing so, maintaining a proper level of abstraction is important, modifying the level of abstraction by minimizing or maximizing artefacts is deemed important. Lastly, functional support to navigate by selecting an artefact type or the relationship between artefacts should be taken into account as well.

6) Define templates – To promote consistency when structuring artefacts, organizations must be able to define and manage templates. Utilization of templates ensures that artefacts are structured consistently. Templates can be required by the machine that is responsible for executing the templates, being business decisions and underlying business logic. Organizations must be able to modify templates to match their context, on top of being able to use standard templates (usually included by the vendor of the software).

C. Specification

With regards to the specification capability, three functional categories were identified: 7) Import Artefact, 8) Export Artefact, and 9) Compare Artefacts.

7) Import Artefact – Similar to the import of sources, import of artefacts is useful as it enables roles to efficiently create or modify artefacts without having to manually insert one of many variables required to do so. Because organizations organize their elicitation and design capabilities different, either supported by information systems or by using word processors, this category requires functional support for different formats or type of documents, see also functional category one. Additionally, when importing artefacts, a role must be able to select what artefacts, type of artefacts and relationships to import. According to the data, a translation of annotation and artefacts between the elicitation and specification capabilities may be required. This means that a role must be able to translate annotations automatically into artefacts utilized in the specification capability. Lastly, because more artefacts are shared nowadays, also between colleague government institutions, import of artefacts from external data sources must be supported.

8) Export Artefact – At some point during or after the specification of business decisions and business logic, a user must be able to export artefacts, which can have several reasons. Usually, this is for either the testing/acceptation, communication or documentation of the business decisions and underlying business logic. Each reason requires different file formats, thus the user must be able to select the type of document that must be exported. Additionally, the representation of the contents in the export is an important factor, depending on the reason for the export. A user must be able to select the representational notation in which the contents are presented in the exported document, i.e., decision tables [36], structured English (controlled natural language) [37] or The Decision Model (model-based) [16]. Similar to importing artefacts, a user must be able to modify whether all artefacts within a given scope or a selection of artefacts or artefact types are exported.

9) Compare Artefacts – The comparison of artefacts is different from the comparison of sources as it focuses on artefacts that are internally created, modified or implemented. Comparison of artefacts must be supported in an automated way in which the machine recognizes and labels Create, Update, and Delete modification types. While comparing artefacts, presentation of meta-data of the artefacts is important, as well as it allows for quick identification and reduces human error.

D. Overlapping functional categories

With regards to the overlapping functional categories that show overlap with all three capabilities, six functional categories were identified: 10) Verify Artefact and Relations, 11) Capture Artefact Meta-data, 12) Capture Additional Artefact Information, 13) Create Overviews, 14) Filter Artefacts, and 15) Capture Artefact Relationships.

10) Verify Artefact and Relations – During the execution of processes along with the elicitation, design and specification capabilities, a multitude of artefacts are created, updated or deleted. The capability following the specification capability is verification, which ensures all business decisions and underlying business logic is syntactically and semantically correct. However, there is no fixed sequentially of the processes conducted as part of the specification or verification capabilities, mainly because this is dependent on how verification is executed, as well as the tooling that is used. Verification can be performed using four techniques: 1) manual detection, 2) manual preventions, 3) automatic detection, and 4) automatic prevention [38]. The data shows that a user must be able to request verification or an artefact or a relationship between artefacts while using a system, as well as being supported by a system that interrupts a user when a syntax or semantic error is detected. Therefore, functional support for a combination of automatic detection (initiated by a user) and automatic prevention must be taken into account.

11) Capture Artefact Meta-data – This functional category focuses on all data captured to support the governance capability, which consists of three sub-capabilities: 1) traceability management, 2) version management, and 3) validity management. More meta-data captured in the elicitation, design and specification capabilities result in more efficient and effective governance during the entire lifecycle of a business decision and its underlying business logic. For example, development status is more efficiently determined when all artefacts under a business decision that is being designed and specified are accompanied by a status and/or version number, which is required for proper version management. For validity management, this means that a user must be able to capture and store variables that represent the validity status of the artefact as provided by the source. For traceability management, this means that a user must be able to capture and store variables that focus on coupling artefacts with each other in a specific format. Additionally, the organization must be able to modify the functionality to capture meta-data as the requirements with regards to governance are different for each organization.

12) Capture Additional Artefact Information – Additional to capturing meta-data required for the governance capability, the data shows a demand for functional support for capturing additional artefact information that is not required to be able to execute or govern the business decision and underlying business logic. Additional artefact information is required, mainly, due to two reasons. First, it enables more effective communication among stakeholders that are responsible for (parts of) the artefacts being created or modified. Second, it enables communication with end-users or clients actually using the business decision, i.e., a governmental portal in which citizens apply for child benefits. The variables allowed to be additionally captured with regards to an artefact depends on the organization and its context. Examples of additional artefact information that were identified in the data are: explanations, motivation/rationale, notes, design or specification decisions per person or role, help text or appendices. Also, a user must be able to capture additional artefact information in each of the, usually, several abstraction levels, i.e., fact-level, decision logic-level, and decision requirements-level [6].

13) Create Overviews – In most organizations large amounts of artefacts are utilized to implement business decisions and underlying business logic. These amounts can pose challenges when searching or reporting certain artefacts, artefact relationships or artefact types. A user must, therefore, be able to create overviews (also referred to as reports) per artefact or other units of analysis. One type of overview that is often identified in the data are meta-data overviews (i.e., generating an overview with all version numbers and validity periods of an artefact), which emphasize that there must be functional support to create overviews for meta-data as well. Additionally, similar to exporting artefacts, a user must be able to select the representational notation in which the contents of the overview are presented. Lastly, depending on the type of modification that has to be processed regarding an implemented business decision and its underlying business logic, users must be able to find and replace efficiently within such overviews.

14) Filter Artefacts – Additional to searching certain artefacts, artefact relationships or artefact types, our data shows that filtering and sorting functionality is deemed important. Additionally, filtering or sorting is not only required for certain artefacts, artefact relationships or artefact types, but meta-data as well.

15) Capture Artefact Relationships – Relationships between artefacts are essential to create decompositions, as well as to ground traceability. Therefore, a user must be able to capture relationships between artefacts, on all abstraction levels of business decisions and business logic. Additionally, organizations must be able to modify relationship types to match their context, on top of being able to use standard relationship types (usually included by the vendor of the software).

With regards to the overlapping functional categories that show overlap with the Design and Specification capabilities, two functional categories were identified: 16) Define Artefact and 17) Issue Management.

16) Define Artefact – According to the data, artefacts that comprise a business decision and underlying business logic are created in the design and specification capabilities. All organizations utilize different stakeholders and tooling. Therefore, a user must be able to define artefacts in multiple representational notations, such as mentioned under functional category export artefact. Another measure to improve efficiency when defining artefacts is to re-use existing artefacts, while a user must be able to change all variables of the existing artefact. Because artefacts are often created or modified by more than one role, collaboration could improve when there is functional support for simultaneously working on artefacts. Additionally, transparent presentation to see which stakeholders have the responsibility and who is working on a (part of a) artefact, should be supported.

17) Issue Management – Collaboration between stakeholders during the development of business decisions and business logic poses several communication challenges. To mitigate this, functional support for issue management is required. Issue management should enable the registration of issues to be solved per artefact in each abstraction layer. Furthermore, all stakeholders must be able to maintain a to-do list, also with the goal to effectively balance the work between relevant stakeholders.

With regards to the overlapping functional categories that show overlap with the elicitation and specification capabilities, one functional category was identified: 18) Artefact Change Support.

18) Artefact Change Support – Changes to sources impacting business decisions and underlying business logic are inevitable, as well as errors that force the organization to modify artefacts throughout the elicitation and specification processes. While we argue that Artefact Change Support could be of importance as a functionality for the design capability, our data did not contain requirements aimed towards the need for artefact change support in the design processes.

The required collaboration between stakeholders or individuals sharing role responsibilities to modify business decisions and underlying business logic often includes hierarchy. For example, based on experience level, some roles or individuals are allowed to process a modification but are disallowed to process the actual change. Functional support to approve changes is deemed important and should be taken into account. Similarly, roles or individuals tasked with reviewing changes made should be supported to roll-back these changes, for example, when errors are detected. Meta-data is an important factor to be taken into account when processing changes but requires additional labour to maintain manually for each change. Therefore, a user must be supported by automatically modifying the meta-data of the changed artefact or suggesting changes to the meta-data so that the user can approve them.

VI. CONCLUSIONS

The goal of this research is to derive a functional architecture that other organizations could utilize to design BRM solutions. To be able to do so, the following research question was addressed in this paper: "Which functional requirement categories should be taken into account when designing a BRM functional architecture for the elicitation, design and specification capabilities?" In order to answer this question, we utilized case study research and conducted three rounds of coding, involving 536 functional requirements specified by seven large Dutch governmental agencies. From a theoretical perspective, our study provides a fundament for future research towards (functional) architecture development in the BRM research field. This is needed as the current knowledge base lacks empirically grounded research into the functional application that facilitate the implementation of BRM capabilities at organizations. From a practical perspective, (governmental) organizations, can use the architectural views per BRM capability presented in this paper as guidance. Organizations that are innovating by applying automating products and services with business decisions and business logic are often searching for guardrails to design their BRM solutions. The results in this paper offer an empirically grounded functional view, based on a large collection of functional requirements, which could function as a guardrail.

VII. DISCUSSION AND FUTURE RESEARCH

Like in this study, the conclusions are solely drawn based on data collected from seven Dutch governmental institutions, which limits the generalizability of the results presented. The first limitation is the sampling, which prevents broad generalization towards other industries. However, we argue that the goal of the functional architectures is to represent a guardrail to be used as a best practice, organizations active in industries other than the government, can utilize what fits best with their context. Also, the sample size is limited and a broad generalization of the results can be achieved when larger sample sizes are used to collect and validate the data, as well as validate the functional architecture. Future research should, therefore, focus on incorporating larger amounts of functional requirements, preferably from a mix of different industries to further validate the current set of functional requirement categories, as well as to compare between different industries with the goal to provide situational sets of functional requirements. This enables better contextualization of the functional architectures based on the industry and organization using the functional architectures.

To create a functional architecture covering all BRM capabilities mentioned in the introduction of this paper, more research is needed. This is necessary as business decisions and business logic are processed in and by several other BRM-related processes and stakeholders before being implemented. Furthermore, as can be derived from Table 2, one organization submitted secondary data which comprised no functional requirements according to our coding but contained functional requirements for other BRM capabilities outside the scope of this paper. For transparency, we retained the organization in the data collection.

Another limitation is the lack of a mixed-method approach to construct the functional architectures. While literature analysis, case study research and secondary data analysis is combined during this research, future research should focus to further improve upon the validity and generalizability of the research results by executing a mixed-method approach. Doing so also enables the inclusion of more data and wider validation of results due to the quantitative viewpoint of the mixed-method approach. Such an approach would also ensure a solid means to validate the functional architecture presented in this paper.

REFERENCES

[1] M. W. Blenko, M. C. Mankins, and P. Rogers, "The Decision-Driven Organization," *Harv. Bus. Rev.*, vol. 88, no. 6, pp. 54–62, Jun. 2010.

[2] P. Rogers and M. Blenko, "Who has the D?," *Harv. Bus. Rev.*, vol. 84, no. 1, pp. 52–61, 2006.

[3] J. Boyer and H. Mili, *Agile business rule development: Process, Architecture and JRules Examples.* Springer Berlin Heidelberg, 2011.

[4] M. L. Nelson, R. L. Rariden, and R. Sen, "A lifecycle approach towards business rules management," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, 2008, pp. 113–123.

[5] S. Schlosser, E. Baghi, B. Otto, and H. Oesterle, "Toward a Functional Reference Model for Business Rules Management," in *2014 47th Hawaii International Conference on System Sciences*, 2014, pp. 3837–3846.

[6] K. Smit and M. Zoet, "An Organizational Capability and Resource Based Perspective on Business Rules Management," in *Proceedings of the International Conference On Information Systems (ICIS)*, 2018, pp. 1–17.

[7] M. L. Nelson, J. Peterson, R. L. Rariden, and R. Sen, "Transitioning to a business rule management service model: Case studies from the property and casualty insurance industry," *Inf. Manag.*, vol. 47, no. 1, pp. 30–41, Jan. 2010.

[8] A. Kovacic, "Business renovation: business rules (still)

the missing link," *Bus. Process Manag. J.*, vol. 10, no. 2, pp. 158–170, 2004.

[9] D. Arnott and G. Pervan, "A critical analysis of decision support systems research," *J. Inf. Technol.*, vol. 20, no. 2, pp. 67–87, 2005.

[10] D. Arnott and G. Pervan, "A critical analysis of decision support systems research revisited: the rise of design science," *J. Inf. Technol.*, vol. 29, no. 4, pp. 269–293, Dec. 2014.

[11] M. Zoet, *Methods and Concepts for Business Rules Management*, 1st ed. Utrecht: Hogeschool Utrecht, 2014.

[12] K. Smit, M. Zoet, and M. Berkhout, "Functional Requirements for Business Rules Management Systems," in *Proceedings of the 23rd Americas Conference on Information Systems (AMCIS)*, 2017.

[13] T. Morgan, *Business rules and information systems: aligning IT with business goals*. Addison-Wesley Professional, 2002.

[14] Object Management Group, "Decision Model And Notation (DMN), Version 1.1," 2016.

[15] the Open Group, "ArchiMate® 3.0 Specification," 2016.

[16] B. Von Halle and L. Goldberg, *The Decision Model: A Business Logic Framework Linking Business and Technology*. CRC Press, 2009.

[17] K. Smit and M. Zoet, "Management Control System for Business Rules Management," *Int. J. Adv. Syst. Meas.*, vol. 9, no. 3–4, pp. 210–219, 2016.

[18] I. Graham, *Business rules management and service oriented architecture: a pattern language*. John wiley & sons, 2007.

[19] S. Liao, "Expert system methodologies and applications—a decade review from 1995 to 2004," *Expert Syst. Appl.*, vol. 28, no. 1, pp. 93–103, Jan. 2004.

[20] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, no. 3. SEI Series in Soft. Eng., 2012.

[21] M. A. Babar and I. Gorton, *Proceedings of the 4th European Conference on Software Architecture (ECSA)*. Copenhagen, Denmark: Springer, 2010.

[22] N. Bateman, "Operating model: an exploration of the concept," Loughborough, UK, 2017.

[23] A. Murphy, J. Kirwin, and A. K. Razak, "Operating models - Delivering on strategy and optimizing processes," 2016.

[24] M. B. Kelly, "The telemanagement forum's enhanced telecom operations map (eTOM)," *J. Netw. Syst. Manag.*, vol. 11, no. 1, pp. 109–119, 2003.

[25] J. Huschens and M. Rumpold-Preining, "IBM Insurance Application Architecture (IAA) — An overview of the Insurance Business Architecture," in *Handbook on Architectures of Information Systems*, Berlin/Heidelberg: Springer-Verlag, 1998, pp. 669–692.

[26] M. Fischbach, T. Puschmann, and R. Alt, "Service-Lifecycle-Management," *Bus. Inf. Syst. Eng.*, vol. 5, no. 1, pp. 45–49, Feb. 2013.

[27] J. Schön, E. M. Thomaschewski and M. J. Escalona, "Agile Requirements Engineering: A systematic literature review," *Comput. Stand. Interfaces*, vol. 49, no. 1, pp. 79–91, 2017.

[28] M. Cohn, *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.

[29] R. K. Yin, *Case study research: Design and methods*. Sage publications, 2013.

[30] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, 2009.

[31] A. Strauss and J. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 3rd ed., vol. 3. Thousand Oaks, CA: SAGE Publications Ltd., 2015.

[32] H. E. Tinsley and D. J. Weiss, "Interrater Reliability and Agreement," in *Handbook of Applied Multivariate Statistics and Mathematical Modeling*, San Diego, CA: Academic Press, 2000, pp. 95–124.

[33] A. Böhm, B. Glaser, and A. Strauss, "Theoretical Coding: Text Analysis in Grounded Theory," *A Companion to Qual. Res.*, pp. 270–275, 2004.

[34] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice Third Edition*, 3rd ed. Addison-Wesley, 2012.

[35] D. M. Buede and W. D. Miller, *The engineering design of systems: models and methods*. John Wiley & Sons, 2016.

[36] J. Vanthienen, "Ruling the business: about business rules and decision tables.," in *New Directions in Software Engineering*, 2001, pp. 103–120.

[37] Business Rule Solutions, "Rulespeak," 2017. [Online]. Available: http://www.rulespeak.com/nl/. [Accessed: 19-March-2020].

[38] K. Smit, J. Versendaal, and M. Zoet, "Identifying Challenges in BRM Implementations Regarding the Verification and Validation Capabilities at Governmental Institutions," in *PACIS 2017 Proceedings*, 2017.

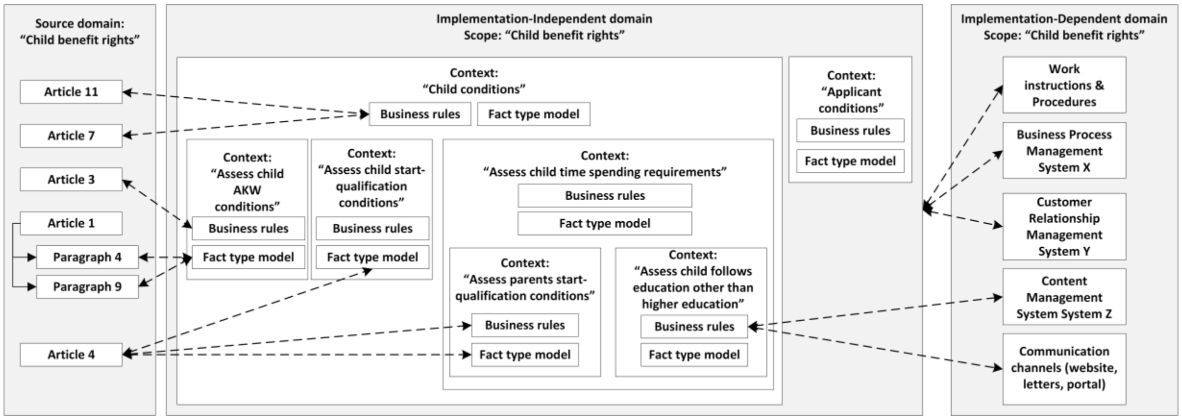

Figure 2. Example of a business decision with underlying business decisions and business logic

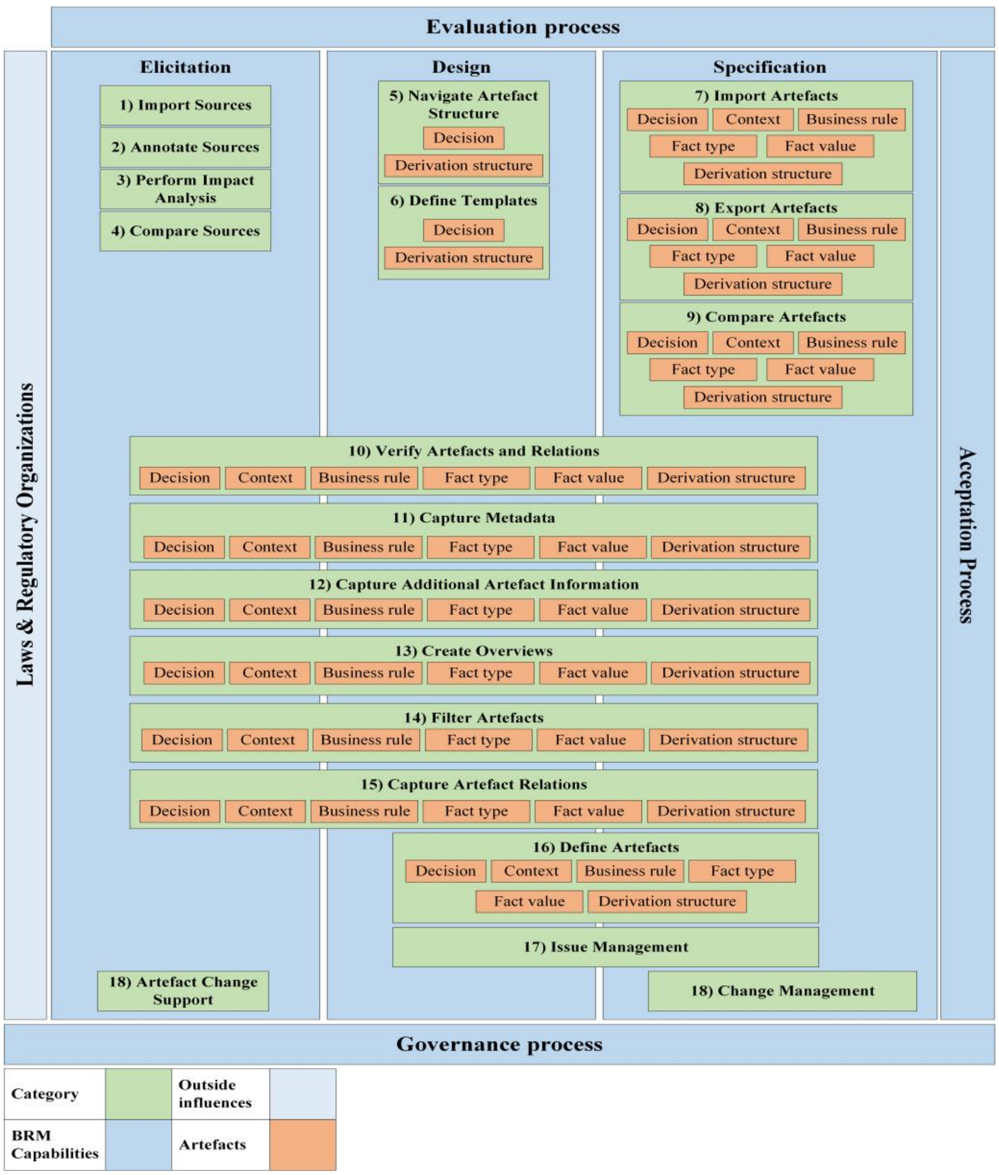

Figure 3. BRM Functional Architecture for the Elicitation, Design and Specification capabilities