TECHNISCHE UNIVERSITÄT DRESDEN

RBPMN: A ROLE-BASED BPMN FOR INTEGRATING STRUCTURE AND BEHAVIOR MODELS

VON TAREK SKOUTI

Preamble

The versatility of roles has for too long been overlooked. Roles have been applied in different fields to model context-related information, but in an isolated way. The application of roles consistently at all abstraction levels (concept, languages, applications, and software system) can lead to context-sensitive systems. For that reason, the DFG funds a research training group on "RoSI - Role-based software infrastructure for continuous context-sensitive systems" (GRK 1907 https://wwwdb.inf.tu-dresden.de/grk/). The author of this paper is part of the research training group and researches a role-based integration of structure and behavior modeling in his dissertation.

This work aims to introduce a role-based approach integrating structural and behavioral modeling and designing and developing a role-based process modeling language that does not force practitioners to learn a new modeling language. A role-based extension for the industry standard of process modeling BPMN was developed to ensure practicality.

This paper is the result of the first year of work.

Contents

List of figures

List of tables

1 Introduction

Business processes (BPs) are modeled to achieve business goals. BP models are used to analyze the as-is BP, to gain insights on weaknesses and impact of the BP, to redesign the BP, and at the end, to implement the BP (Gonzalez-Huerta et al., 2017). There are various BP modeling languages (BPML) such as UML Activity Diagrams, BPMN, Event-driven Process Chains, Role-Activity-Diagrams, Role-Interaction-Networks, and YAWL. BPMN has been established as the de-facto standard BPML (Zarour et al., 2019). Using an established modeling language allows model exchange and communication between domain analysts and software developers. A BP that has been modeled is continuously under review. The 8th survey of the series "The state of business process management" shows that 9 out of 10 companies are conducting BP improvement projects. BP initiatives that focus on BP redesign occur in almost half of the enterprises (Harmon & Garcia, o. J.).

BPs are continuously changed and adapted due to more demanding customers, optimizations, changes in regulations, legal and compliance requirements, or new technologies. This results in new BP models and new implementations in software systems to adhere to the new requirements. Usually, much time passes until the software system can adequately support the adapted process, resulting in higher costs. First, the BP model is changed by the domain analyst. In the second step, the BP model is used to specify software system requirements to be implemented by software developers in the third step. However, BPMN is neither natively executable nor does it allow the expression of structural elements. Thus, a standard BPMN model first must be mapped to a model closer to implementation like UML class diagrams. Since BPMN models are graph-structured and UML class diagrams are block-structured, naturally, there is a mismatch (Ouyang et al., 2009). How to overcome this mismatch is a research subject that continuously provides new approaches. One approach proposed by various researchers (Balabko et al., 2004; Kemp et al., 2019; Saidani & Nurcan, 2006; von Rosing et al., 2017) is role-orientation. BPMN is not yet role-oriented, but how a role-based BPMN (RBPMN) can overcome this mismatch and other challenges is presented in this paper. Overall, we contribute a role-based modeling approach that better integrates structure and behavior models, solves current modeling challenges, and is explicitly suited for digitalized BPs (including Artificial Intelligence (AI)), enterprise transformation, and context-aware systems.

2 Background

BPMN 2.0 is maintained by the Object Management Group (OMG) and was introduced in 2011 (OMG, 2011). With the 2.0 release, an extension mechanism by addition was presented. If a BPMN extension adheres to the extension mechanism, it can be integrated seamlessly. As there is no official guideline on how a BPMN extension is designed, some extensions do not adhere to the extension mechanism. The systematic literature reviews of Braun and Esswein (2014) and Zarour et al. (2019) show that many extensions, primarily if the extension is not focused on a single domain, are not following the extension mechanism. Stroppi et al. (2011) proposed an extension mechanism based on UML profiles that can be used for domain-specific extensions.

To understand the specifics of BP (behavior) models, we interpret BPs to consist of activities performed in a sequence or parallel by multiple performers in relationships under a specific context for a business goal.

Saidani and Nurcan (2006) distinguish three types of BP activities: Manual activities, system activities, and user interaction activities. Manual activities can be performed without the aid of a software system. System activities are fully automated tasks that are performed by the underlying software system. A user interaction activity is performed in conjunction with a software system. The context of a BP specifies the conditions that must be met to achieve the business goal. Without a context, activities are not goal-oriented (Santra & Sankhayan, 2018). The analysis of a BP can be made in terms of *roles* played by *performers* and performed activities. We follow the definition of roles proposed by Schön et al. (2019): "A role is a contextual modeling construct with state and behavior that is fulfilled by an object or its roles to represent it in the user's context and extend or change its corresponding specification and interactions." The application of roles in BP modeling has been proposed in various ways (Balabko et al., 2004; Caetano et al., 2005; Kemp et al., 2019; Meertens et al., 2010; Saidani & Nurcan, 2006; von Rosing et al., 2017). Roles are not unique to behavior modeling. Roles have been applied in structural modeling in the form of the business role object specification (BROS) (Schön et al., 2019), programming languages in the form of SCROLL (Kühn, 2017), and security in the form of role-based access control (Ferraiolo & Kuhn, 1992), to name a few areas. One of the challenges the widespread use of roles still faces is the manifoldness of interpretations of the role concept, which hinders exploiting their full potential (Kühn, 2017; Zhu & Zhou, 2008).

Figure 1 illustrates the potential of roles and how roles support the transformation of enterprise-models. Using a role-based structural modeling language such as BROS, an enterprise behavior model can be derived from its structural model. The behavior model provides the behavior of the
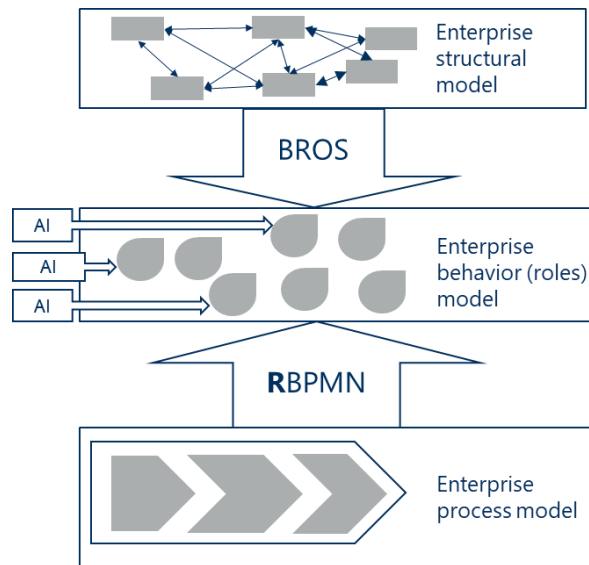

Figure 1: Roles supporting enterprise transformation

roles, which is needed to determine responsibilities and capabilities. Using a role-based BP modeling language provides the relationships and interactions between roles and relates them to

business activities and goals. Also, Artificial Intelligence (AI) may transform the behavior of roles and interactions between them. An intelligent system, which is an information system that uses AI, does not only support roles. It can play roles, act as a performer, and transform other physical entities to act as new performers, thus creating new roles (Zimmermann et al., 2020). A role-based process model (RBPMN) specifying a system can be mapped to a role-based structural model (BROS) without information loss and facilitates implementation and maintenance. The resulting role model *"permits researchers and software designers to use role models to communicate their ideas, domain models, and software designs"* (Kühn, 2017).

3   Business Process Modeling Challenges

BP modeling is the activity performed during BP identification, the starting point of the BP management (BPM) cycle. The result is an "as-is" BP model, which can identify weaknesses, resource constraints, and formulate improvement opportunities. The improvement is presented as a "to-be" BP model. The "to-be" BP model is then used to derive software requirements to support the "to-be" BP (Recker & Mendling, 2016). Deriving software requirements is not a trivial task, and as BPs are becoming more complex, the complexity of the software systems supporting the BPs increases. It is not only the BP that is becoming more complex; performers' responsibility in a BP increases. Intelligent systems do not only support human performers. The physical components that intelligent systems extend become performers themselves. AI can adapt a BP to different situations based on the context information provided (Koehler, 2018; Zimmermann et al., 2020). We identified three main challenges that BP modeling faces: The variety of performers, adaptability, and context-awareness.

3.1   Variety of Performers

Many human performers are turning into knowledge workers. A knowledge worker is a person that has a multitude of different tasks and performs activities in different roles and responsibilities. However, the knowledge worker has poorly been addressed in the various BP modeling approaches. The rising number of knowledge workers requires businesses to be more flexible and variable in their BPs. Customers are becoming more demanding in terms of the necessary flexibility of the business, too. Therefore, knowledge workers and intelligent systems must play multiple roles in BPs (von Rosing et al., 2017). The rise of the knowledge workers and digitalization require a semantic change of the term performer. AI can support humans at each BP step separately and eventually replaces humans in BPs. The performer is not just humans anymore (Friedl, 2018; Koehler, 2018). Therefore, the number and variety of roles a performer (human, machine, or data) plays in a BP increase. A performer must perform compliance checks, auditing, monitoring, accounting, billing, and more. Especially in small companies, this is not done by different performers for each activity, but by the same performer in various roles. The challenge "variety of performers" is not just modeling more performers with more activities, but also separating the activity responsibility from the performer. If the specified performer is not available, a BP should not come to a stop. If the responsibility is attached to the role instead of the performer, it can be

fulfilled by another performer. The second problem has been addressed in various ways (Charfi et al., 2010), and the use of roles to achieve separation of responsibility and separation of duty appears to be a promising approach (Bera et al., 2018; Kemp et al., 2019; Saidani & Nurcan, 2006).

3.2   Adaptability

Customer demands for more variations and higher flexibility in BPs increase the need for software and information systems' *adaptability*. The same holds for models, which specify the BPs and the system. Modeling variation within a BP model is often targeted by BPMN extensions (Zarour et al., 2019). Modeling languages are an important instrument for introducing adaptability. One approach to provide adaptability in models is modularity. However, there are multiple conceptual models within an enterprise regarding organizational structure, BPs, software systems, and security. A common concept is needed to allow for modularity, separation of concerns, and the inclusion of intelligent systems and emerging performers in each model.

Adaptation strategies can be implemented in advance for expected situations, but they are crucial in unexpected situations. The ever-changing external and internal environment requires continuous adaptation of the BP at execution time and not just at design time (Santra & Sankhayan, 2018). An adaptable BP model would allow for the predefined application of changes during the BP execution, but this remains an open challenge (Cognini et al., 2014). An adapted BP must still adhere to compliance and other regulations. Adapting the BP is just part one; adapting the system supporting the BP in a timely manner is part two of the adaptability challenges. Providing support for more process variations results in more complex systems. Complex BPs with many performers and many activities require more complex systems to facilitate support. Changes to BPs and the system supporting these BPs happen continuously, shown by the amount of process reengineering projects in companies (Harmon & Garcia, o. J.). The challenge of adaptability is modeling adaptable BPs and expressing adaptability to develop maintainable systems. One approach that has been used in various fields to introduce adaptability is roles (Cognini et al., 2014; Jamel et al., 2018; Schön et al., 2019). Roles as a common approach in modeling and programming languages create higher interconnectivity between the various conceptual models. Roles could solve the adaptability problem.

3.3   Context-Awareness

Both linking performers to roles as well as BP adaptation depend on contextual information. Therefore, context-awareness presents the third main challenge. Dealing with context-awareness has been the focus of many BPM researchers (Jamel et al., 2018; Saidani & Nurcan, 2009; Santra & Sankhayan, 2018). The challenge of context-awareness can be split into different problems that must be solved first. The first problem is that "context" has multiple definitions. Stemming from linguistics, the Cambridge English Dictionary defines it as the "cause of events", "related events", and "surrounding words" (*Context Meaning in the Cambridge English Dictionary*, 2020). In BPM research, the definition of context is not singular either. Dey (2001) defines context as "any information that can be used to characterize the situation of entities that are considered relevant to

the interaction between a user and an application, including the user and the application themselves". Born et al. (2009) define context as "the environment in which a BP artifact is used". Santra and Sankhayan, (2018) recently defined BP context as "the information that impacts the design and execution of a BP either externally or internally". There are more definitions of context, and each varies slightly from the other. Therefore, each approach aiming at solving context-awareness must define the context.

The second problem is *awareness*. Context is sometimes static, sometimes dynamic. It can change rapidly, and thus, it is essential to consider various contextual information to adapt the BP while maintaining integrity. The execution of a BP is often context-dependent, and so are the characteristics of entities involved in the BP. Decisions can be made differently based on the context, while the data provided stays the same. To deal with context changes and to use new context information to adapt a BP is the challenge of context-awareness. One approach that we have seen mentioned in BP modeling is again role-orientation (Kemp et al., 2019; Saidani & Nurcan, 2009). Incorporating role-orientation into BPMN, which is not natively context-aware, is a promising approach to solve this challenge.

4   Proposing Roles as a Solution

The research stream "role-orientation" is gaining momentum. The importance of roles cannot be overstated, but their representation in conceptual modeling is unsatisfactory. Section 2 presented the application of roles in conceptual models. In the process modeling language BPMN, some features of roles are realized in the form of swimlanes. Swimlanes, however, do not cover the full functionality of roles (Bera et al., 2018).

The following example of Kühn et al. (2014) presents the key concepts of role-orientation *role, performer and context*. A "*Customer*" role cannot be played without a "*Seller*" role in the context of a "*Shop*". The performers of "*Customer*" and "*Seller*" exist without the "*Shop*" but not in these roles.

From the example, multiple role features can be derived. Roles are played by a performer and do not exist without context. Roles communicate with one another. Roles capture the behavior of a performer. Roles are played by a performer and bestow functionality, responsibility, and characteristics onto the performer. Roles are acquired and removed.

Roles separate responsibility from the performer. Performer P1 is no longer specified as responsible for Activity A, but instead, the Role R holds the responsibility. If performer P1 is unavailable to perform activity A1, performer P2 playing Role R can perform activity A1 without deviating from the BP model, the BP is adapted. The performer playing the role is not fixed to the role, nor is the role fixed to the performer. Instead of performers, assigning roles to business goals increases flexibility, as the performer who plays the role can switch.

Roles solve the problem of separation of duty. Separation of duty, e.g., is used to minimize fraud. This is achieved by dividing duty for an activity into multiple mutually exclusive roles. A performer cannot create and audit a document. However, a performer can perform both activities

creation of a document and audit of a document. It must be ensured that a performer cannot do both activities for the same document. Role-orientation provides a role prohibition feature to ensure mutually exclusive roles (Saidani & Nurcan, 2006).

The variety of performers that partake in BPs increases, but the roles played in BPs do not change at the same rate. Different performers perform the same activities in different contexts. However, the performers are playing the same role. The complexity each performer offers to the process can be decreased by focusing on their role. New performers that do not fit to play any current role will eventually add complexity. The number of roles in an enterprise is smaller than the number of performers.

Roles allow adaptability, as their existence is context-dependent. Roles can perform activities. This allows role-activity assignments. An activity is always assigned to a business goal. Based on the activity assignments, the essential roles for a BP can be derived. All possible BP activities can be derived based on the roles involved, allowing to adapt BP execution (Saidani & Nurcan, 2006; Schön et al., 2019). The adaptation is critical in case of unexpected situations that cannot be modeled beforehand. Further, the performer playing a role can change without disrupting the BP.

Roles are context-sensitive, as they only exist and interact in a context. We follow Dey's definition of context (Dey, 2001). A role-based BP model is, by nature, always context-sensitive. Context must be modeled as it is obligatory for roles. Roles are context-aware as their behavior, and their properties change if the context differs. Therefore, a role-based BP model can provide guidelines for creating context-aware systems, especially using role-based programming languages (Kühn, 2017).

The business process modeling challenges can be solved using roles, but roles' applicability does not end there. As explained in section 3, Roles are a common concept in different conceptual models found in enterprises. All the different models like security, behavior, structure are needed to understand an enterprise fully. Either an integrated or a non-integrated enterprise modeling approach is chosen. The situation of non-integrated models in enterprises is depicted in figure 2. Each model has its connections and associations to other model concepts. Behavior models describe which data needs to be accessed. Security models describe which data can be accessed by whom, but not in which context. The models are consistent by themselves but not integrated (Vernadat, 2020). Knowledge is lost in translating concepts between the different models, and change in one model results in more changes in other models. A change in the behavior model must be communicated to security and data structure and implemented respectively. Additional coordination between data structure and security might be needed to ensure the behavior change is fully implemented. The changes in data structure might require new security changes. Suppose more conceptual models are taken into account. In that case, each change in one model increases the system's complexity as a whole until the interconnections between the models become less apparent.

We follow the idea of Drouot & Champeau (2019) and Werner et al. (2018) that roles should be used consistently in each model and provide the common concept needed for better integration. Figure 3 depicts such an integrated role-based approach, where roles are found in each model.

Role-based access control for security, role data and structure models like BROS for structure and role process models for behavior. Changes in one model will not result in a manifold of changes in other models. Instead, only the role is changed without increasing the overall complexity of the system.
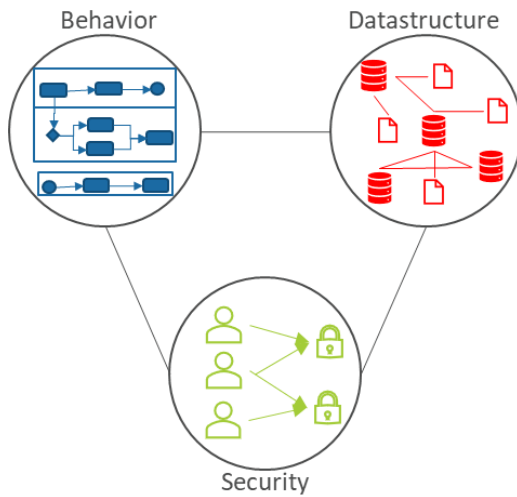

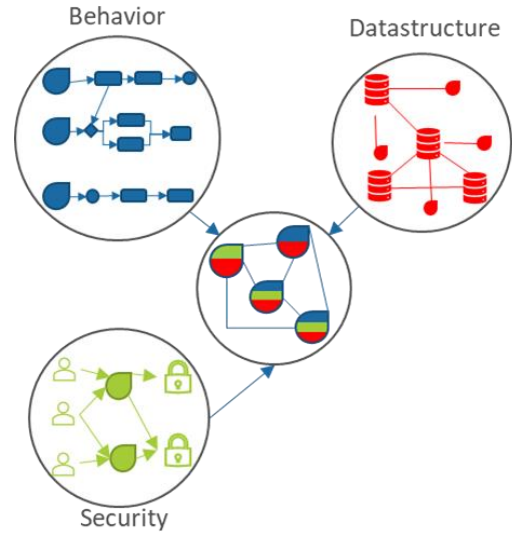Figure 2: Connection between conceptual models

Figure 3: Roles as a common concept in different conceptual models

5   The Role-based Business Process Model and Notation

Designing a solution to the three challenges of business process modeling is the goal of this work. Roles, as a proposed solution for each challenge, are a promising approach. Therefore, based on these findings, we developed a role-based BPMN extension called RBPMN (= Role-based Business Process Model and Notation). We based the design artifact's development (a language extension) on the design science method (Hevner, 2007).

We have presented three of the main challenges that BP modeling faces today. RBPMN is built on ideas and approaches that researchers have presented (Caetano et al., 2005; Meertens et al., 2010; Saidani & Nurcan, 2006; von Rosing et al., 2017). To ensure the application of role-based approaches, we decided to extend the BP modeling language BPMN, which is already established in businesses and research (Geiger et al., 2018), instead of developing another language. RBPMN's role features realization is analyzed, and a first demonstration of the artifact is presented in an example.

RBPMN can express system requirements for adaptable, context-aware, and maintainable systems. An RBPMN model serves as a construction plan for the development of adaptable and maintainable software systems. RBPMN is a role-based extension of the BPMN language that allows incorporating the multitude of performers, separation of duty, increases model reuse, and offers help in the system implementation by providing an integrated approach of behavior and structural modeling.

5.1 BPMN Extension Mechanism

BPMN2.0 provides an extension mechanism by addition to add non-standard elements or artifacts. The extended BPMN diagram should look like a BPMN element, so that knowledge about BPMN should be sufficient to understand the extended BPMN diagram. The BPMN Core flow elements (Events, Activities, and Gateways) must not be changed. A BPMN extension is compliant with the extension mechanism if it follows specified principles (OMG, 2011). 1. The extension must not change the shape of a graphical element or marker. 2. The line style of a graphical element must not be changed to conflict with any other element. 3. An added shape must not conflict with any other BPMN element or marker. 4. The added graphical element or marker is either used to highlight a specific attribute of a BPMN element or represent a new subtype.

Figure 4 shows how the RBPMN elements have been integrated into BPMN. Figure 4 does not show all BPMN elements. Elements that do not have a relation to RBPMN elements were there-
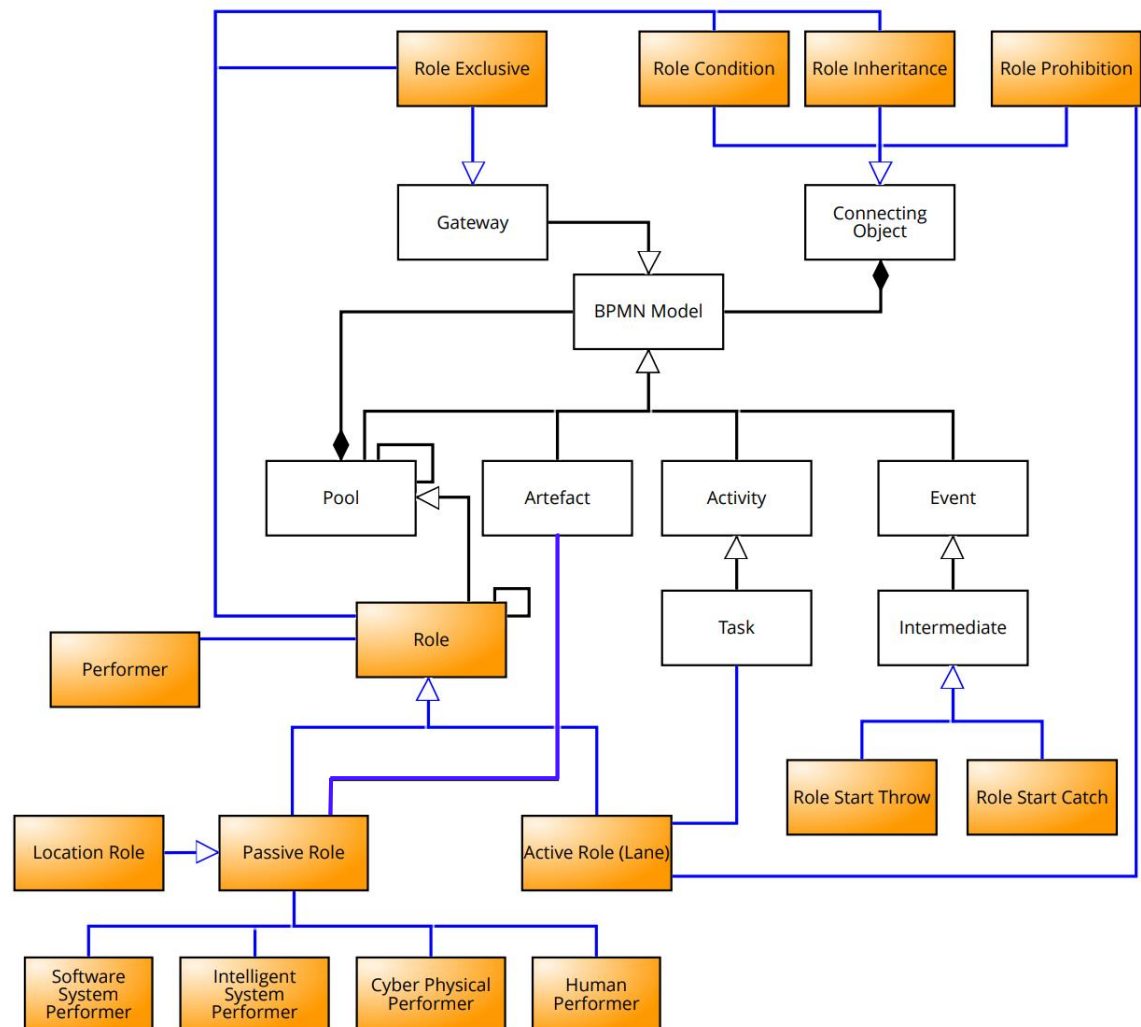

Figure 4: RBPMN Extension element relationships to BPMN elements

fore excluded from the figure. The new elements and how these elements are associated with each other and existing BPMN elements are highlighted by color. All but one added element (Performer) are subtypes of existing elements. The performer element highlights a specific attribute.

The existence of roles requires pool and performer. Activities and connecting objects are related to the role, not the performer. An activity is always related to a role, and a role can have multiple activities. The new elements are not in conflict with the existing BPMN elements described in detail in the subsequent chapters.

5.2 Composition of Roles in a Business Process

In BPMN, a pool or swimlane represents the responsibilities of activities. It is up to the modeler to decide what the lane represents (role, participant, system, department, enterprise) (OMG, 2011). This leads to confusion when models are interchanged or read by people with a different concept in mind when talking about lanes. Organizations, participants, or systems can play roles. Humans, physical entities, software, and intelligent systems, to name some examples, can play active and passive roles. For humans, it could be a physician (active) or a patient (passive) in the context of an examination. The OMG (2011) specified systems as candidates for the swimlane, and, therefore, systems are possible performers for an active role in RBPMN. When it is briefly used as support in a BP for a single activity or the activity does not require AI, such as a simple, repetitive activity, it is specified as a passive role. Intelligent systems perform multiple complex activities. With AI tools, the intelligent system can self-adapt and change its behavior based on context information (Koehler, 2018). We consider these as potential performers for both active and passive roles. The data object and data store remain and are used the same way they were used before. They now describe the role played by the performer of the data object or data store.

5.2.1 Active Roles

We propose to define active roles as what is expressed as a swimlane in BPMN. The expressiveness is maintained, and the meaning of the swimlanes is exact. However, swimlanes themselves are not enough to express binding or separation of duties or resource assignment constraints for non-human performers (Bera et al., 2018; Cabanillas et al., 2015). Active roles allow the integration of deep roles, which means that a role can play a role instead of a performer playing a role. The BPMN element subprocess links two processes in a hierarchical order. Subprocesses restrict the number of participants in it to the swimlane it is modeled in the superprocess. Deep roles allow the inclusion of more roles and separate concerns in the subprocess itself.

5.2.2 Passive Roles

Not every role in BP performs activities. Nonetheless, the entities playing the specified roles are necessary to achieve the business goal of BP. We define these roles as "passive roles". Performers of passive roles can express more than the BPMN artefacts data object or data store. Any business object could potentially play a passive role. A unique passive role is the location role for BPs, where the location is mandatory to achieve the business goal, which is explained in detail later.

5.3   Connector Elements

The existing connector elements are insufficient to express role prohibition between active roles and to ensure compliance. The separation of responsibility is not expressible. Furthermore, inheritance between passive roles and the conditional existence of roles cannot be expressed.

5.3.1   Role Prohibition

As its name suggests, the role prohibition connector prohibits two active roles from being played by the same performer in an instance. The connected roles are mutually exclusive. For passive roles, we propose using different shades to express that different performers must play these roles.

5.3.2   Role Inheritance

There are two types of inheritance, one for each role-type. The BPMN2.0 Element subswimlane expresses inheritance for active roles. A subrole contains the properties and behavior of the role it is located in and extends these further. Inheritance for passive roles, which were newly introduced in this work or data objects in general, cannot be expressed in BPMN. Only Activity, Event, and Sequence Flow are the elements that can work as a source or target of a data association, but there is no inheritance between passive roles. To solve this problem, we propose a new connector. The role inheritance connector expresses that the performer previously playing the supertype role is now playing the inherited role. This also allows for the modeling of evolution during a BP.

5.3.3   Role Condition

The role condition provides that event A or activity A must have happened for role B to exist. If the activity is obsolete in the process instance, so is the role in this instance.

5.4   Location Role

The location role is a unique passive role. It can span over multiple pools within a pool over multiple activities of different roles or only over a single task of a single role. It is used to express that the activities included must happen in the same specified location. A location performer can be a laboratory, a pure air room, an IT system, and more. There can be multiple location roles in a BP, which allows for the modeling of different contexts. For example, a conference can be held online, offline, or hybrid, with only some audience being physically present. The location role performer can change, and the role can be played by a physical and a virtual performer. This allows for adapting to different situations more quickly.

5.5   Role Start Event

A **role-start-catch** and a **role-start-throwing** event are added. These events are used to express that a performer acquires another role without abandoning the role it currently plays.

5.6   Gateways

A role exclusive gateway expresses if splitting the sequence flow that different performers must play the roles in the split paths. The necessity becomes apparent in the following situation. Depending on the path followed, a customer (role) is either played by a company or a person (performer). Gateways in BPMN always provide a splitting and a merging ability. When a role exclusive gateway is used to merge sequence flows of multiple roles simultaneously played by a single performer, the performer keeps one role after the merge and abandons all other roles.

5.7   RBPMN Syntax

The concrete syntax to express the introduced elements is shown in Table 1. The use of icons for passive roles is proposed to distinguish between different performer types. The table is not exhaustive, as there are more performer types. To denote that the same performer must play different roles, we propose the use of patterning. Patterning is a tool of expressiveness that is not fully used in BPMN. It is allowing entities to be seen in conjunction or separation (Kummer et al., 2016). The use of color instead of patterning is possible, but not always necessarily the best choice and depends on the domain analyst and developer [33].

Table 1: RBPMN Extension Elements

Element	Expression	Symbol
Passive Role (Human Performer)	An additional human performer is required for the activity	
Passive Role (Software System Performer)	A software system is required for the activity	
Passive Role (Intelligent System Performer)	An intelligent system is required for the activity	
Passive Role (Cyber-Physical Performer)	A cyber-physical object is required for the activity	
Location Role	Activities must happen in the same location (e.g., room, information system)	
Active Role Prohibition	The same performer cannot play Role A and Role B. (Role Prohibition)	
Passive Role Inheritance	Role A is a generalization of Role B	
Role Condition	Activity A (source) must have been executed for Role B (target) to exist	
Role Start Throwing Event	Event signaling that a performer acquired another role to play simultaneously	
Role Start Catching Event	Event signaling the role the performer acquired	
Role Exclusive Gateway	When splitting the sequence flow, the possible flows cannot be performed by the same performer. When merging paths, a performer playing more than one role must abandon all but one role after the gateway.	

5.8   **Process Role Hierarchy**

The BPMN specification provides no guidelines for a good modeling style. Practitioners have, therefore, taken the lead in developing style guides for good modeling practice. Enterprises offering BPMN solutions like Camunda and Signavio provide their modeling guidelines in the form of model best practices (*BPMN-Richtlinien | BPMN modeling guidelines*, o. J.). One particular modeling style, "process hierarchy", must be readdressed if roles are involved. BPMN is used to model processes on the management and operative level. Management level processes often contain subprocesses of the operative level, which in turn can again contain subprocesses. A pool or lane that contains a subprocess must be modeled in the subprocess itself and must not be changed. RBPMN modeling style guidelines make an addition at this point. While active roles and pools must not be changed in subprocesses according to BPMN modeling guidelines, new passive roles can be added to subprocesses in RBPMN.

6   **Role-Feature Integration**

The need for a role-based business process modeling language has been presented in section 2. The choice for a BPMN extension instead of a newly developed BPML was presented in section 4. The newly developed RBPMN adheres to the BPMN extension mechanism (compare figure 2) as no BPMN element has been changed. The added elements can, therefore, be integrated into existing BPMN modeling solutions. Steimann (2000) and Kühn (2017) found in total 27 role features. A fundamental addition of Kühn (2017) to the role features is Compartments. The term Compartment was introduced in Kühn et al. (2014) to summarize the terms Environments, Institutions, Teams, and Ensembles, which were all used to describe context without using the term context. Kühn et al. (2014) defined Compartment as an: "objectified collaboration with a limited number of participating roles and a fixed scope" (Kühn et al., 2014, p. 146). If compared to the pool element of BPMN, we find similarities between both concepts. A pool acts as the container for lanes, and the sequence flows between activities. The sequence flows cannot cross the boundaries of a pool (OMG, 2011). A pool represents an internal process of a process participant expressing the specified participant's internal business objective. The modeler limits the number of participating active and passive roles in a pool. The scope of the sequence flow in a pool is fixed. The high similarity of BPMN's concept pool and compartment of role-orientation leads us to interpret pools as compartments.

The RBPMN was designed with the idea to realize these features in the industry-standard BPMN. Not all the role features apply to the model level (M1). Some are only applicable to the runtime level (M0). RBPMN's realization or not realization of the role features that apply to the model level is described in detail in table 3. For better understanding within the context of BPMN modeling, we decided to rename the terms object and compartment in Table 2 to performer and pool in table 3.

Table 2: Friedrich Steimann's 15 and Thomas Kühn's 12 additional classifying features extracted from (Kühn, 2017)

1. Roles have properties and behaviors	(M1, M0)
2. Roles depend on relationships	(M1)
3. Objects may play different roles simultaneously	(M1, M0)
4. Objects may play the same role (type) several times	(M0)
5. Objects may acquire and abandon roles dynamically	(M0)
6. The sequence of role acquisition and removal may be restricted	(M1, M0)
7. Unrelated objects can play the same role	(M1)
8. Roles can play roles	(M1, M0)
9. Roles can be transferred between objects	(M0)
10. The state of an object can be role-specific	(M0)
11. Features of an object can be role-specific	(M1)
12. Roles restrict access	(M0)
13. Different roles may share structure and behavior	(M1)
14. An object and its roles share identity	(M0)
15. An object and its roles have different identities	(M0)
16. Relationships between roles can be constrained	(M1)
17. There may be constraints between relationships	(M1)
18. Roles can be grouped and constrained together	(M1)
19. Roles depend on compartments	(M1, M0)
20. Compartments have properties and behaviors	(M1, M0)
21. A role can be part of several compartments	(M1, M0)
22. Compartments may play roles like objects	(M1, M0)
23. Compartments may play roles which are part of themselves	(M1, M0)
24. Compartments can contain other compartments	(M1, M0)
25. Different compartments may share structure and behavior	(M1)
26. Compartments have their own identity	(M0)
27. The number of roles occurring in a compartment can be constrained	(M1)

Table 3: Role feature realization in RBPMN

Realization: Fully ■   Partially ◩   Not realized ☐   Only instance-level Ø

Role Feature	Realization	
1	RBPMN was designed to integrate structure and behavior using roles. The behavior of a role is modeled by the activities associated with the role, and the properties are expressed inside the role like BPMN.	■
2	No process in RBPMN exists without at least two roles that interact with each other. A role's existence depends on the possibility to interact with another role.	■
3	Performer and role are in a m:n relationship. Performers can play different roles in the process simultaneously. There are two ways to model that a performer of an active role plays another active role simultaneously. One is the use of subroles if the roles share structure. The other is the use of the role start event. Passive roles that are played simultaneously are modeled using the same shading without a role inheritance connection.	■
4	Not applicable to model level	Ø
5	Not applicable to model level	Ø
6	The sequence of role acquisition and removal follows the sequence flow. There is no additional restriction.	◩
7	Passive role performers can play the same role without being related to each other.	■

8	The BPMN 2.0 standard provides this possibility for *active roles* using subswimlanes. RBPMN, as an extension of BPMN, realizes this feature by nature. The role inheritance connector expresses a passive role playing another passive role.	■
9	Not applicable to model level	Ø
10	Not applicable to model level	Ø
11	The features of a performer are specific to the role it plays.	◪
12	Not applicable to model level	Ø
13	Behavior is expressed as the sum of activities, associations, and relationships a role has. There is no restriction on two roles sharing the same behavior.	■
14	Not applicable to model level	Ø
15	Not applicable to model level	Ø
16	The relationship between roles depends on the modeled sequence flow and associations. There is no additional constraint element	□
17	The relationships are constrained at design time.	□
18	Roles are grouped within their pool.	■
19	Roles are only modeled within pools; thus, their dependence on pools is given by design.	■
20	The pool's properties and behavior is the sum of all activities and associations modeled within the pool.	■
21	Active roles cannot be expressed as part of several pools in RBPMN. Passive roles used in several pools are modeled outside of pools and connected with the data association to the respective pools.	□
22	Pools can play active roles like performers, but not passive roles. A collapsed pool plays a role like a performer	■
23	Pools can be modeled as roles within themselves under certain conditions over multiple connected models. The activity subprocess modeled in an active role connects two models in hierarchical order. The pool in model 1, where the subprocess activity A is modeled, can play a role in the subprocess model 2.	◪
24	Pools can be modeled within a pool in the BPMN2.0. Pools can contain pools in RBPMN.	■
25	Only pools within the expanded pool can be modeled to share structure and behavior. The BPMN modeling guidelines suggested having only one white box pool, and pools outside of it are modeled as black boxes; thus, their behavior is not modeled. While different pools can share structure and behavior, it is typically not modeled outside of an expanded pool.	◪
26	Not applicable to model level	Ø
27	The number of roles occurring in a pool is defined at design time. There is no additional constraint expressible.	■

Most role-features that apply to the model level are realized in RBPMN. Some tradeoff between role-features and BPMN extensibility must be made. One development goal of RBPMN was to adhere to the BPMN extension mechanism. Thus, the tradeoff decisions were made one-sided in favor of achieving a standardized BPMN extension instead of achieving full role-feature coverage. We find that some role-features were already introduced in BPMN, which might not have been intentional.

7   Modeling Case Study

The following figure 3 presents a simplified paper submission process modeled with RBPMN. The BP model in figure 3 is an extended version of the BP model presented in (Kirchberg et al., 2009). The BP starts with the "Call for Papers" (CfP) and has multiple possible endings. In the best-case scenario, the paper is submitted, reviewed, reworked, and funding is obtained to participate in the conference, thus ending the BP. Other endings are possible, like the researcher not

being interested in the CfP, a rejection decision from the reviewers, or no funding is obtained. The active roles are *Researcher*, *Reviewer*, *Track Chair*, and *Author* with three subroles *Author with Submission*, *Author with Review*, and *Author of accepted Paper*. The passive roles are *RPA* (played by an *intelligent system performer*) and *Draft* with the three subroles *Submitted Paper*, *Paper with Review,* and *Accepted Paper*.

It is often the case that authors with a submission are asked to review other submissions. Therefore, the performer of the role author can play the role of reviewer or track chair, but the performer of the role author cannot play reviewer for their paper. This rule is specified externally, but not in the BP instance itself in traditional BPMN (OMG, 2011). Role-prohibition expresses this rule in the BP model. It is found between reviewer and author. Subroles inherit the properties of their parent roles. The three subroles of author are also mutually exclusive to the reviewer in a BP instance.

The passive roles in the example are not all mutually exclusive. The intelligent system performer plays the passive role of Robotics Process Automation (RPA) and performs a plagiarism check supporting the Track Chair. RPA is currently used for repetitive work, mimicking humans (van der Aalst et al., 2018). The role RPA starts existing if a paper is submitted, shown by the role condition.

It is expressed that the same data performer plays four different roles "Draft", "Submitted Paper", "Paper with Review", and "Accepted Paper" in RBPMN by using the same pattern. In standard BPMN, there would be one data object to represent the paper with connections to many different activities and each active role. This can lead to confusion about the state of the performer "Paper". Therefore, a separation of the state and the performer is necessary. This can be solved using Roles. The concrete syntax shows the passive role ("Draft") as written in the performer symbol (Data Object). The same pattern is used for all passive roles of the same performer. This expresses that the same performer must play these four roles.

The passive role Submitted Paper is different from Draft by being blinded and having a submission ID but inherits the content of Draft. The Paper with Review role inherits the Submission ID and has added reviews. The Accepted Paper role has the Authors, Acknowledgements, and the final layout. The passive roles inherit information and characteristics from the other passive roles, and this inheritance is shown using the role-inheritance.

The location role spans over multiple activities and roles. The submission and review activity must happen in a single system to ensure that role-prohibition is being upheld. In the example, the location role is an intelligent system, given that a RPA role exists in it. Some activities are performed outside the location role like "obtain funding".

The example shows that the BPMN language is not inherently changed. The added elements fit comfortably and are clearing up the confusion. The variety of performers (active and passive) is taken care of. The adaptability is increased. The context-awareness is provided by understanding

that the role-prohibition is only set for a single BP instance. The performer playing the role author can still play reviewer, but the separation of duty is maintained.
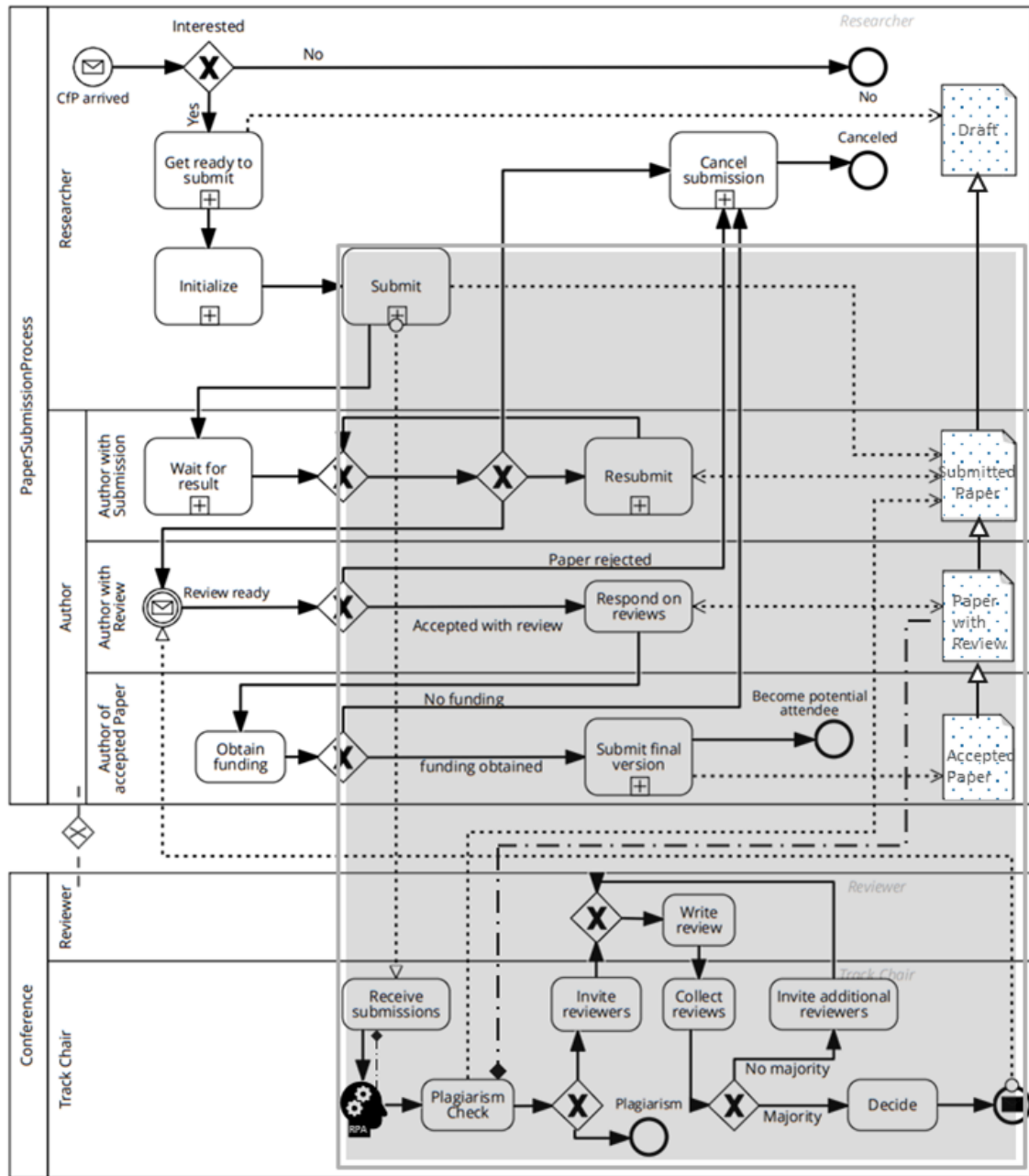
Figure 5: RBPMN-Model Paper-Submission-Process

8   Discussion

The challenges mentioned in section three are addressed in other research streams as well. Therefore, RBPMN is put into perspective to approaches from these streams.

One research stream is aspect-orientation. Aspect-orientation was introduced to target insufficiencies of object-oriented programming. The three core concepts are join points, pointcuts, and advice. Join points represent execution points of a program that can be selected individually and as

a set using pointcuts. The advice is several join points interacting and depending on one an-other. Aspect as the core element of aspect-orientation is a set of several pointcuts and advice. It was introduced into BPMN with the extension AO4BPMN to target the separation of concerns over multiple BPs as one of the BP modeling challenges (Charfi et al., 2010). A BP is not inde-pendent of other BPs, and the same or similar tasks can be found in different BPs. The following example explains the necessity of the separation of concerns. A compliance task in BP "Model A" could be different from a compliance task in BP "Model B" but is addressed in both models as "check compliance". To express that compliance tasks are the same, AO4BPMN uses BPMN activities and events to model join points. Pointcuts are introduced as a new graphic element in the form of Data Objects with an associated annotation "pointcut". The advice is modeled as a BPMN subprocess with a text annotation "Advice". Aspects are modeled using Pools and Swim-lanes with the text annotation "aspect". AO4BPMN was improved to AOBPMN, AO-BPM, and AO-BPM2.0 (Carvalho et al., 2018). The latest improvement, AO-BPM2.0, splits BP and con-cern. The total number of elements needed to model a BP increases resulting in a more complex model. All improvements so far did not conform to the BPMN Metamodel, as stated in a recent Systematic Literature Review (Zarour et al., 2019).

Another research stream is subject-orientation, which originated from the distributed software field. The subject is the key concept that is modeled. It represents a BP related functionality, while the actor is the instance of it. This results in a better modeling concept for communications and parallel activities if more than one subject is involved. Any BP involving more than one subject is suited for it. Collaborations are by nature multi-actor systems and can, therefore, be modeled subject-oriented. The BP modeling language S-BPM was introduced to express Subject-Orienta-tion. It can be mapped to the industry-standard BPMN and vice versa, but both lose some of their representational strength (Singer, 2019).

C-BPMN is a BPMN extension that focuses on context-awareness and adaptability. C-BPMN models create an additional level surrounding the BP model to incorporate adaptability by mod-eling additional scenarios (Santra & Sankhayan, 2018).

RBPMN does not change the BPMN modeling language drastically. The added elements increase the expressiveness of the BPMN. The research stream role-orientation benefits from the role-based process modeling. We claim that RBPMN, together with role-based structural models, can cope with enterprise transformation and future challenges. However, this remains to be shown in our future research.

9   Conclusion

The presented RBPMN was designed to accommodate modeling challenges. RBPMN solves the presented challenges: variety of performers, adaptability, and context-awareness. Roles are a common concept in behavioral and structural modeling of systems applied in software develop-ment. Roles offer a seamless, architecturally clean mechanism to introduce AI into BP models. A role-based BP model based on RBPMN allows changes of performers (from human to AI) without redesign of the BP. Roles are context-aware by their nature.

RBPMN provides a new approach to derive and design software artifacts without changing the BPMN modeling language. The new elements, active and passive roles, and the new connectors prohibition, condition, conjunction, and inheritance are seamlessly integrated into BPMN.

We see two main *contributions* of our approach. The first being that the challenges listed in sec. 3 are solved. The variety of performers is considered by the variety of roles expressible (active and passive roles). Responsibility is attached to the role, not the performer. Software developers benefit significantly from RBPMN, especially in a role-based environment due to the adaptation possibilities that roles offer. Software developers are supported in the implementation of future-proof, context-aware systems. Software requirements in terms of relationships, interactions, and security are more easily derived from a role-based model. The second contribution is the increased mapping possibility from process and behavior modeling to structural modeling with roles as the integrating concept. The target group of the RBPMN extension are both domain analysts and software developers but for different reasons. As it has been presented in this paper, Roles are a concept that is ubiquitous throughout the information technology domain. Roles, as a shared concept in process modeling (RBPMN), structural modeling (BROS) (Schön et al., 2019), software development (CROM and SCROLL) (Kühn, 2017), and security (RBAC) (Ferraiolo & Kuhn, 1992), reduce the information loss during translation of one field to another. Domain analysts can thereby express unique information using roles without increasing the complexity of the model. Because RBPMN adheres to the BPMN extension mechanism other BPMN extension mechanism conform extensions such as extensions for the healthcare domain can be included in RBPMN. The holistic role-based approach has not reached its full potential. The full potential of RBPMN can be achieved in a role-based environment. In future work, the RBPMN must be validated in terms of applicability, usefulness, and maintainability of the resulting software artifacts.

10 Acknowledgments

References

Balabko, P., Wegmann, A., Ruppen, A., & Clément, N. (2004). The Value of Roles in Modeling Business Processes. *CAiSE Workshops*, 9.

Bera, P., Burton-Jones, A., & Wand, Y. (2018). Improving the representation of roles in conceptual modeling: Theory, method, and evidence. *Requirements Engineering*, *23*(4), 465–491. https://doi.org/10.1007/s00766-017-0275-9

Born, M., Kirchner, J., & Müller, J. P. (2009). Context-driven Business Process Modelling: *Proceedings of the Joint Workshop on Advanced Technologies and Techniques for Enterprise Information Systems*, 17–26. https://doi.org/10.5220/0002201500170026

BPMN-Richtlinien | BPMN modeling guidelines. (o. J.). Abgerufen 9. Dezember 2020, von https://www.modeling-guidelines.org/de/guidelines/

Braun, R., & Esswein, W. (2014). Classification of Domain-Specific BPMN Extensions. In U. Frank, P. Loucopoulos, Ó. Pastor, & I. Petrounias (Hrsg.), *The Practice of Enterprise Modeling* (Bd. 197, S. 42–57). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-45501-2_4

Cabanillas, C., Knuplesch, D., Resinas, M., Reichert, M., Mendling, J., & Ruiz-Cortés, A. (2015). RALph: A Graphical Notation for Resource Assignments in Business Processes. In J. Zdravkovic, M. Kirikova, & P. Johannesson (Hrsg.), *Advanced Information Systems Engineering* (Bd. 9097, S. 53–68). Springer International Publishing. https://doi.org/10.1007/978-3-319-19069-3_4

Caetano, A., Zacarias, M., Silva, A. R., & Tribolet, J. (2005). A Role-Based Framework for Business Process Modeling. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 13c–13c. https://doi.org/10.1109/HICSS.2005.49

Carvalho, L. P., Cappelli, C., & Santoro, F. M. (2018). AO-BPM 2.0: Aspect Oriented Business Process Modeling. In E. Teniente & M. Weidlich (Hrsg.), *Business Process Management Workshops* (Bd. 308, S. 719–731). Springer International Publishing. https://doi.org/10.1007/978-3-319-74030-0_57

Charfi, A., Müller, H., & Mezini, M. (2010). Aspect-Oriented Business Process Modeling with AO4BPMN. In T. Kühne, B. Selic, M.-P. Gervais, & F. Terrier (Hrsg.), *Modelling Foundations and Applications* (Bd. 6138, S. 48–61). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13595-8_6

Cognini, R., Corradini, F., Gnesi, S., Polini, A., & Re, B. (2014). Research challenges in business process adaptability. *Proceedings of the 29th Annual ACM Symposium on Applied Computing - SAC '14*, 1049–1054. https://doi.org/10.1145/2554850.2555055

Dey, A. K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing*, *5*(1), 4–7. https://doi.org/10.1007/s007790170019

Dictionary, O. (2020). dictionary.cambridge.org. https://dictionary.cambridge.org/dictionary/english/context

Drouot, B., & Champeau, J. (2019). Model Federation based on Role Modeling: *Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development*, 72–83. https://doi.org/10.5220/0007363500720083

Ferraiolo, D. F., & Kuhn, D. R. (1992). Role-Based Access Controls. *15th National Computer Security Conference*, 554–563.

Friedl, A. (2018). Meeting Industrie 4.0 Challenges with S-BPM. *Proceedings of the 10th International Conference on Subject-Oriented Business Process Management - S-BPM One '18*, 1–6. https://doi.org/10.1145/3178248.3178249

Geiger, M., Harrer, S., Lenhard, J., & Wirtz, G. (2018). BPMN 2.0: The state of support and implementation. *Future Generation Computer Systems*, *80*, 250–262. https://doi.org/10.1016/j.future.2017.01.006

Gonzalez-Huerta, J., Boubaker, A., & Mili, H. (2017). A Business Process Re-Engineering Approach to Transform BPMN Models to Software Artifacts. In E. Aïmeur, U. Ruhi, & M. Weiss (Hrsg.), *E-Technologies: Embracing the Internet of Things* (Bd. 289, S. 170–184). Springer International Publishing. https://doi.org/10.1007/978-3-319-59041-7_10

Harmon, P., & Garcia, J. (o. J.). *The State of Business Process Management—2020* (Survey Nr. 8; The State of Business Process Management). https://www.bptrends.com/bptrends-state-of-business-process-management-2020-report/

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems*, *19*(2), 87–92.

Jamel, L., Saidani, O., & Nurcan, S. (2018). Flexibility in Business Process Modeling to Deal with Context-Awareness in Business Process Reengineering Projects. In J. Gulden, I. Reinhartz-Berger, R. Schmidt, S. Guerreiro, W. Guédria, & P. Bera (Hrsg.), *Enterprise, Business-Process and Information Systems Modeling* (Bd. 318, S. 35–48). Springer International Publishing. https://doi.org/10.1007/978-3-319-91704-7_3

Kemp, N., Arzumanyan, M., Zachman, J. A., & Skurzak, K. (2019). The organization of the future with Role Oriented Process Modelling. In *The complete Business Process Handbook: Extended Business Process Management* (Bd. 2). Morgan Kaufmann.

Kirchberg, M., Sorensen, O., & Thalheim, B. (2009). A BPMN Case Study: Paper Review and Submission System. In *GI Jahrestagung*.

Koehler, J. (2018). Business Process Innovation with Artificial Intelligence: Levering Benefits and Controlling Operational Risks. *European Business & Management*, *4*(2), 55. https://doi.org/10.11648/j.ebm.20180402.12

Kühn, T. (2017). *A Family of Role-Based Languages* [Monograph, TU Dresden]. https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-228027

Kühn, T., Leuthäuser, M., Götz, S., Seidl, C., & A\s smann, U. (2014). A metamodel family for role-based modeling and programming languages. In *Software Language Engineering* (S. 141–160). Springer. http://link.springer.com/chapter/10.1007/978-3-319-11245-9_8

Kummer, T.-F., Recker, J., & Mendling, J. (2016). Enhancing understandability of process models through cultural-dependent color adjustments. *Decision Support Systems*, *87*, 1–12. https://doi.org/10.1016/j.dss.2016.04.004

Meertens, L. O., Iacob, M. E., & Nieuwenhuis, L. J. M. (2010). Towards a Definition of Role-Related Concepts for Business Modeling. *2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops*, 138–146. https://doi.org/10.1109/EDOCW.2010.23

OMG (Hrsg.). (2011). *Business Process Model and Notation (BPMN), Version 2.0.* https://www.omg.org/spec/BPMN/2.0

Ouyang, C., Dumas, M., Aalst, W. M. P. V. D., Hofstede, A. H. M. T., & Mendling, J. (2009). From business process models to process-oriented software systems. *ACM Transactions on Software Engineering and Methodology*, *19*(1), 1–37. https://doi.org/10.1145/1555392.1555395

Recker, J., & Mendling, J. (2016). The State of the Art of Business Process Management Research as Published in the BPM Conference: Recommendations for Progressing the Field. *Business & Information Systems Engineering*, *58*(1), 55–72. https://doi.org/10.1007/s12599-015-0411-3

Saidani, O., & Nurcan, S. (2006). A Role-Based Approach for Modelling Flexible Business Processes. *Proceedings of the CAISE*06 Workshop on Business Process Modelling, Development, and Support*, 111–120.

Saidani, O., & Nurcan, S. (2009). Context-awareness for adequate business process modelling. *2009 Third International Conference on Research Challenges in Information Science*, 177–186. https://doi.org/10.1109/RCIS.2009.5089281

Santra, D., & Sankhayan, C. (2018). C-BPMN: A Context Aware BPMN for Modeling Complex Business Process. *CoRR*, *abs/1806.01333*. arXiv. http://arxiv.org/abs/1806.01333

Schön, H., Strahringer, S., Furrer, F. J., & Kühn, T. (2019). Business Role-Object Specification: A Language for Behavior-aware Structural Modeling of Business Objects. In T. Ludwig & V. Pipek (Hrsg.), *Wirtschaftsinformatik 2019 Proceedings* (S. 244–258). https://aisel.aisnet.org/wi2019/

Singer, R. (2019). An Ontological Analysis of Business Process Modeling and Execution. *ArXiv:1905.00499 [Cs]*. http://arxiv.org/abs/1905.00499

Steimann, F. (2000). On the representation of roles in object-oriented and conceptual modelling. *Data & Knowledge Engineering*, *35*(1), 83–106. https://doi.org/10.1016/S0169-023X(00)00023-9

Stroppi, L. J. R., Chiotti, O., & Villarreal, P. D. (2011). Extending BPMN 2.0: Method and Tool Support. In R. Dijkman, J. Hofstetter, & J. Koehler (Hrsg.), *Business Process Model and Notation* (Bd. 95, S. 59–73). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-25160-3_5

van der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2018). Robotic Process Automation. *Business & Information Systems Engineering*, *60*(4), 269–272. https://doi.org/10.1007/s12599-018-0542-4

Vernadat, F. (2020). Enterprise modelling: Research review and outlook. *Computers in Industry*, *122*, 103265. https://doi.org/10.1016/j.compind.2020.103265

von Rosing, M., Arzumanyan, M., & Zachman Sr., J. A. (2017). The Relationship Between Ontology and Modelling Concepts: Example Role Oriented Modelling. *International Journal of Conceptual Structures and Smart Applications*, *5*(1), 25–47. https://doi.org/10.4018/IJCSSA.2017010102

Werner, C., & Aßmann, U. (2018). *Model Synchronization with the Role-oriented Single Underlying Model*. 10.

Zarour, K., Benmerzoug, D., Guermouche, N., & Drira, K. (2019). A systematic literature review on BPMN extensions. *Business Process Management Journal*. https://doi.org/10.1108/BPMJ-01-2019-0040

Zhu, H., & Zhou, M. (2008). Roles in Information Systems: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(3), 377–396. https://doi.org/10.1109/TSMCC.2008.919168

Zimmermann, A., Schmidt, R., Jugel, D., & Möhring, M. (2020). Evolution of Enterprise Architecture for Intelligent Digital Systems. In F. Dalpiaz, J. Zdravkovic, & P. Loucopoulos (Hrsg.), *Research Challenges in Information Science* (S. 145–153). Springer International Publishing.

TECHNISCHE UNIVERSITÄT DRESDEN
FAKULTÄT WIRTSCHAFTSWISSENSCHAFTEN

Bisher erschienene Titel in der Reihe:
Dresdner Beiträge zur Wirtschaftsinformatik

lfd.Nr.	Autor/Autoren	Titel
1/94	Werner Esswein, Eric Schoop, Wolfgang Uhr	Der Studiengang Wirtschaftsinformatik an der Fakultät Wirtschaftswissenschaften der Technischen Universität Dresden
2/94	Eric Schoop, Stefan Papenfuß	Beiträge zum computerunterstützten Lernen
3/94	Werner Esswein, Klaus Körmeier	Führung und Steuerung von Softwareprojekten im Kapsel-Modell
4/94	Werner Esswein	Entwurf integrierter Anwendungssysteme
5/94	Gundula Heinatz	CSCW und Software Engineering
6/94	Marco Lehmann-Waffenschmidt, Klaus-Peter Schulz	Umweltinformationssysteme in der öffentlichen Verwaltung
7/94	Eric Schoop, Stefan Papenfuß, Jan L. Plass	Echolot: Making an Abstract Hypertext Machine Concrete - A client/server architecture for authoring and learning business processes -
8/95	Eric Schoop, Thomas Schraml	Vom Durchfluß- zum Kreislaufbetrieb: Neuorientierung zu einem integrierten Umweltinformationsmanagement
9/95	Gundula Heinatz	COST14-CoTech Project P4 CSCW and Software Engineering Dresden, 22-23 May 95 Meeting Papers
10/95	Eric Schoop, Thomas Schraml	Vorschlag einer hypertext-orientierten Methode für eine strukturierte Umweltberichterstattung und -zertifizierung
11/96	Eric Schoop, Uwe Jäger, Stefan Pabst	Potentiale elektronischer Märkte
12/96	Eric Schoop, Ralph Sonntag, Katrin Strobel, Torsten Förster, Sven Haubold, Berit Jungmann	Vergleichende Übersicht von Angeboten der Bundesländer im Internet
13/96	Eric Schoop, Hagen Malessa, Jan L. Plass, Stefan Papenfuß	Architekturvorschlag für eine offene Hypermedia-Entwicklungsumgebung zur Erstellung verteilter Lernsysteme
14/96	Silvia Brink Wolfgang Uhr	Hypertextbasierte Lernumgebung "Investitionsrechnung" - Konzept und Evaluation -

15/97	Thomas Schraml, Eric Schoop	Umweltinformationsmanagement mit neuen Medien. Elektronische Berichterstattung durch Hypertext-Dokumente
16/97	Eric Schoop	Multimedia-Kommunikation: Chancen für KMU?
17/97	Eric Schoop	Wachstum und Innovation: Herausforderung für ein Informationsmanagement mit neuen Medien
18/97	Ralph Sonntag	Automatisierung der Erstellung und Pflege von Umwelthandbüchern sowie der enthaltenen Verfahrensregelungen
19/97	Eric Schoop	Document Engineering: Methodische Grundlage für ein integriertes Dokumentenmanagement
20/97	Werner Esswein, Gundula Heinatz, Andreas Dietzsch	WISE.xScape - ein Werkzeug zur Unterstützung informeller Kommunikation
21/97	Werner Esswein, Gundula Heinatz	Dokumentation von Leistungs- und Informationsprozessen im Krankenhaus für ein verbessertes Qualitätsmanagement
22/97	Werner Esswein, Gundula Heinatz	Einsatz von technischen Systemen zur Unterstützung von Koordination in Unternehmen
23/97	Werner Esswein, Manuela Räbiger, Achim Selz	Anforderungen an Data-Warehouse-Systeme
24/97	Gerhard Marx	Reaktionsfähigkeitsanalyse (RFA)
25/97	Michael Schaffrath Wolfgang Uhr	Symptomerkennung im Rahmen eines Entscheidungsunterstützungssystems
26/98	Jens-Thorsten Rauer	Strategische Erfolgspotentiale von Informations-infrastrukturen in der deutschen Assekuranz (Individualversicherungsbranchen) - die betriebliche Ressource Informationsverarbeitung
27/98	Stefan Papenfuß	Vorschlag für eine Informationsstruktur
28/98	Eric Schoop	Strukturorientierte Dokumentenmanagement, Aufgaben, Methoden, Standard und Werkzeug
29/98	Ralph Sonntag	Jahresbericht 1994-1997 der Professuren für Wirtschaftsinformatik
30/99	Ina Müller	Integration technologiebezogener Informationen an der TU Dresden in ein Gründerinformationssystem
31/99	Michael Zilker	Einsatz und Nutzenkalküle von Virtual Reality-Projekten in Unternehmensprozessen Auswertung einer Befragung von VR-Anwendern und Ableitung eines Unterstützungsbedarfs durch ein computerbasiertes Beratungssystem
32/00	Andreas Dietzsch Werner Esswein	Modellierung komplexer Verwaltungsprozesse: Arbeitsbericht zum Projekt Finanz 2000
33/01	Sabine Zumpe Werner Esswein	Automatische Unterstützungssysteme für die Steuerberatung
34/01	Jürgen Abrams Wolfgang Uhr	B2B-Marktplätze – Phänomen und organisatorische Implikationen

35/02	Ruben Gersdorf	Verteiltes Content Management für den Document Supply in der Technischen Dokumentation
36/02	Sabine Zumpe Werner Esswein	Konzeptuelle Schnittstellenanalyse von eCommerce Applikationen
37/02	Ernest Kosilek Wolfgang Uhr	Die kommunale elektronische Beschaffung Bericht zum Forschungsprojekt „KeB"
38/02	René Rottleb	„Verzeichnissysteme – ein Stiefkind der Wirtschafts-informatik?"
39/02	Andrea Anders	Ergonomische Dokumente: Wie lassen sich zielgruppen-spezifische Anforderungen an Lesbarkeit und Verständlich-keit sicherstellen?
40/03	Anja Lohse	Integration schwach strukturierter Daten in betriebs-wirt-schaftliche Prozesse am Beispiel des Kundenservice
41/03	Berit Jungmann	Einsatz von XML zur Abbildung von Lerninhalten für E-Learning-Angebote: Standards, Anwendung, Handlungs-bedarf
42/03	Sabine Zumpe, Werner Esswein, Nicole Sunke, Ma-nuela Thiele	Virtuelle B2B-Marktplätze Entstehung, Existenz und Umwandlung
43/04	Torsten Sommer	Modellierung von standardisierten Behandlungsabläufen Begriffsanalyse als Voraussetzung zur Wahl einer geeigne-ten Modellierungssprache
44/04	Eric Schoop	Electronic Business – Herausforderungen im größer ge-wordenen Europa - Proceedings zum Europäischen Integ-rationsforum 2004 an der Fakultät Wirtschafts-wissen-schaften der TU Dresden
45/04	Andreas Hilbert Sascha Raithel	Entwicklung eines Erklärungsmodells der Kundenbindung am Beispiel des High-Involvement-Produktes Automobil
46/04	Andreas Hilbert Sascha Raithel	Empirische Evaluation eines Kausalmodells zur Erklärung der Kundenbindung am Beispiel des High-Involvement-Produktes Automobil
47/05	Sabine Zumpe Werner Esswein Nicole Sunke Manuela Thiele	Die Qualität von Referenzmodellen im E-Commerce
48/05	Daniel Kilper	Ressourcenkritische Parameter XML-basierter Trans-akti-onsstandards in mobilen Datennetzen: Eine Unter-su-chung basierend auf der Informationellen Effizienz nach Hurwicz
49/06	Frank Wenzke	Angebot von Informationen für die Wettbewerberanalyse auf Unternehmenswebsites
50/07	Silke Adam Werner Esswein	Untersuchung von Architekturframeworks zur Strukturie-rung von Unternehmensmodellen

51/07	Markus Westner	Information Systems Offshoring: A Review of the Literature
52/08	Tobias von Martens Andreas Hilbert	Kapazitätssteuerung im Dienstleistungsbereich unter Berücksichtigung des Kundenwertes
53/09	Roy Wendler	Reifegradmodelle für das IT-Projektmanagement
54/10	Christian Leyh Anne Betge Susanne Strahringer	Nutzung von ERP-Systemen und RFIDTechnologie in klein- und mittelständischen Unternehmen - Eine explorative empirische Untersuchung sächsischer KMU
55/10	André Gräning Roy Wendler Christian Leyh Susanne Strahringer	Research about before Research with Standards
56/10		Tagungsband des zwölften interuniversitären Doktorandenseminars
57/10	Andreas Hartmann, Georg Lackermair	Trends im B2C-Online-Handel Ergebnisse einer Expertenbefragung
58/11	Michael Mohaupt, Andreas Hilbert	Systematisierung und Berücksichtigung von Unsicherheiten im Revenue Management
59/11	Christian Leyh, Pia Hübler	Nutzung von ERP-Systemen in sächsischen klein- und mittelständischen Unternehmen Eine explorative empirische Untersuchung
60/11	Christian Leyh, Henrique Gottwald	Nutzung von ERP-Systemen in deutschen klein- und mittelständischen Unternehmen Eine explorative empirische Untersuchung
61/11	René Michel, Igor Schnakenburg, Tobias von Martens, Andreas Hilbert	Effektive Kundenselektion für Vertriebskampagnen auf Basis von Nettoscores
62/11	Christian Leyh	Verwendung von ERP-Systemen im Rahmen der Hochschullehre Auswertung einer Befragung deutscher Universitäten und Fachhochschulen
63/12	Kerstin Förster, Roy Wendler	Theorien und Konzepte zu Agilität in Organisationen
64/12	Tobias Weiß, Andreas Hilbert	Energy Intelligence Anwendung von Business Intelligence auf Daten aus dem Smart Metering

65/12	Christian Leyh, Christian Hennig	ERP – und Campus-Management-Systeme in der Hochschulverwaltung – Ergebnisse einer Befragung deutscher Universitäten und Fachhochschulen
66/13	Tim Pidun	Assessing the Success of Performance Measurement Systems
67/13	Marcus Pfitzner	Visualisierungstrends von Business Intelligence Oberflächen
68/14	Roy Wendler, Theresa Stahlke	What constitutes an agile organization? – Descriptive results of an empirical investigation
69/14	Andreas Schieber, Andreas Hilbert	Entwicklung eines generischen Vorgehensmodells für Text Mining
70/14		Tagungsband des 18. Interuniversitären Doktorandenseminars
71/14	Michael Mohaupt	Konzeptioneller Bezugsrahmen zur Unsicherheitsproblematik im kundenzentrierten Revenue Management
72/21	Tobias Weiß	Die Nutzung von Daten aus dem Smart Metering für innovative Zusatzprodukte - Eine angebots- und nachfrageseitige Analyse im deutschen Markt