

# The Process-Rule Continuum — Can BPMN & SBVR Cope with the Challenge?

Jana Koehler

Lucerne University of Applied Sciences and Arts

Technikumstrasse 21, CH-6048 Horw

Switzerland

jana.koehler@hslu.ch

**Abstract**—With increasing needs for business agility and cost pressures on IT, Business Process Management (BPM) is asked to move towards “Dynamic BPM” and “Intelligent Case Management” instead of freezing process flows in hard-to-change IT solutions. Although business rules are considered an important ingredient of dynamic BPM solutions, only little is understood about the interplay of business processes and business rules.

We report on the results of a case study in the area of electronic billing where we explored the interplay between business processes and business rules based on a set of scenarios for the so-called process-rule continuum proposed by Gartner. We critically review these scenarios and argue that they can be reduced to 4 key patterns of rule usage. We review the BPMN and SBVR standards to which extent they support these 4 key patterns and identify critical gaps that should be addressed by future standardization efforts.

## I. INTRODUCTION

Classical process modeling specifies all the possible paths a process can take for each of its possible instances. For many business processes, this leads to models that are moderate in size and relatively easy to understand, analyze, and implement. However, the growing demand for business agility and the desire to better support and partially automate the knowledge- and communication-intensive processes of knowledge workers brings processes into the focus of attention for which the number of possible paths is dramatically increasing. Not only must more and more complex decisions be captured by a process diagram, but also escalation paths, error handling, and compensation logic should be shown in the model. Furthermore, business rules play an increasingly important role in agile business scenarios where they decide and guard how business (data) objects are handled by a process and how process partners interact.

On the one hand, the Business Process Model and Notation Standard (BPMN) [1], Version 2.0, provides a rich modeling notation for complex processes and process collaboration. It has also added conversation and choreography diagrams to capture how processes (and process partners) interact with each other. On the other hand, the Semantics of Business Vocabulary and Rules Available Specification by the OMG (SBVR) [2] has clarified the semantics of business terms and rules. Unfortunately, both standards recognize the

need for a deeper integration, but at the same time declare it as out of scope.

SBVR discusses the important relationship between processes and rules briefly:

*“... Business Rules deliver factored out, flexible detail to support Business Processes ... Business Processes provide the specific contexts in which Business Rules need to be evaluated ... Business processes are better defined when a business knows where it wants to go (its goals and objectives), and what it needs to do to get there (its strategies, tactics, and policies). Business processes realize the strategies and tactics. Business rules realize the business policies, and both support and constrain the business processes. Business processes, supported by business rules, are associated with organization roles and structure. Some business rules apply to organization structure and roles, independently of processes. There is clearly a need for integration.” [2, p. 391]*

BPMN provides a so-called business rule task to model the static invocation of a business rule calculation in a specific activity of the process:

*“A Business Rule Task provides a mechanism for the Process to provide input to a Business Rules Engine and to get the output of calculations that the Business Rules Engine might provide.” [1, p. 163]*

Business rule tasks in a process model simplify a process model by eliminating alternative process paths from the process and by encoding the handling of different situations into the rules. The input into a rules engine is commonly a business object handled by the process, e.g., an order to be processed or a bill to be issued. The specific values of the object attributes characterize the object instance handled by a specific process instance. These values are analyzed by the business rules and determine which output the rules calculate. The output of the rules is a new value for one or several attributes of the business object that may then influence the further branching of the process. This usage of business rules is well understood today and supported by established business process management suites (BPMS) that often contain rule handling components. BPMS can be extended with dedicated business rule management systems

(BRMS) to better manage and process large and frequently changing rule sets.

## II. SEVEN SCENARIOS IN THE PROCESS - RULE CONTINUUM

Various scenarios of integrating processes and rules can be imagined. A very interesting discussion of the continuum for processes and rules has been published recently in a Gartner report [3]. It proposes seven distinguished scenarios for process-rule integration:

*“One method is to put all work activity, including business rules, in the process itself. The other extreme is a method to create the process dynamically by combining just rules to produce a process. The reality is that there is a continuum of how much activity is put into rules and how activity is put into process. Where to land on that continuum depends on the attributes/goals of process design.” [3, p. 1]*

*Scenario 1: Embedded Rules:* All process paths are directly encoded into the process model. Rules are not explicitly used. Instead, rule-based decisions are encoded in the gateway logic of the process. This pattern is widely practiced today and commonly implemented with the help of a BPMS. It is appropriate for stable processes that do not undergo much change, where change is not time-critical, and the process has very few exceptions. If change is required, the process model and its implementation undergo a classical software development process.

*Scenario 2: Explicit Navigation Rules:* Explicit rules manage and direct the process flows for each process instance. They are separated from the process model and maintained in rule components evaluated by a rules engine. The business process contains business rule tasks that determine when a rule component is executed. This is the second most common scenario today. It is usually implemented using a combined BPMS/BRMS approach. Changing rules is separated from changing the process and takes place by editing and deploying a new set of rules in the BRMS.

*Scenario 3: Complex Navigation/Analysis:* The rules used by the process become more complex and several rule sets are used. Rules take more attributes of the business object(s) into account and apply a more fine-grained distinction of attribute values leading to more fine-grained decisions. Process flows (and results) can be different for each instance. The number of rules is exploding and maintaining a consistent rule set becomes increasingly difficult. Change is required more often, because reactions to specific attribute values must be adjusted more frequently.

*Scenario 4: Rule-Guided Process Behavior:* “In this approach, additional rules, as well as navigation rules, are made explicit, and the rules are packaged to be coordinated to influence process outcomes.” [3, p. 5] The interplay of different business processes and their rule sets to govern a process is an important characteristic of this scenario. A

process does not only have its own complex navigation rules that analyze the business objects and the process context from the perspective of the current process instance, but the process interacts with rules used by other processes that observe much wider business conditions and that can be linked to optimization algorithms. These rules can for example evaluate monitoring results from several previous process instances and feed their evaluations into the decisions of a currently running instance. For example in contrast to Scenario 3, where the navigation rules look only at the purchases of the customer in the running process instance, a Scenario 4 process instance can consult rules that give insights into the monthly sales to a certain group of customers and use these insights in a decision related to the current customer.

*Scenario 5: Rule-Driven Process Composites:* Rules direct the dynamic composition of processes from predefined (process) components. The composition depends on the context of an individual process instance and happens at runtime. No predefined process paths can be drawn in a model. Instead, business rules decide how a process must evolve to contribute towards a larger business goal.

*Scenario 6: Rule-Driven Services:* In contrast to the process components of Scenario 5, which are usually hardened processes, the services of Scenario 6 are “fluid and flexible” [3, p. 6], i.e., they are more fine-grained and scoped to a specific business capability or situation. The orchestration of these services is driven by various rule packages that are bundled together to constitute a business solution. Change takes place at the level of rule packages and service components and due to their fine-grained structure may be required more often and be driven by advanced analytics of monitored component configurations and sophisticated optimization techniques.

*Scenario 7: Fully Rule-Dynamic:* “This is the lowest level of granularity where the processes dynamically configure themselves for every process instance and are controlled more by constraints (guardrails).” [3, p. 6] The key difference of this scenario compared to Scenarios 5 and 6 is the self-changing rule sets, which dynamically adjust to changing conditions at runtime without human intervention. This requires learning capabilities to yield modified or new rules, whereas in all other scenarios, rule sets are defined at design time and do not change at run time.

In the following, we summarize the results of a case study in which billing scenarios from the telecommunications industry were modeled and analyzed in each scenario. Based on this analysis, we argue that the seven scenarios can be reduced to four key patterns of rule usage in processes. Furthermore, we investigate how the BPMN and SBVR standards support each pattern and we identify gaps in the standards. Note that we have to assume readers to be familiar with BPMN and SBVR due to space restrictions. However, we will briefly introduce some key concepts when needed.

### III. CUSTOMER BILLING SCENARIOS

Customer billing has undergone a dramatic change over the last years. In past times, the pricing of consumer goods and services was usually quite simple—a good would cost the same for each customer independent of the point in time when it was purchased. The corresponding billing process thus consisted of a few simple and predefined activities that obtained the purchases of the customer, calculated the price, created the bill, and mailed it to the customer. Figure 1 shows this process in BPMN notation. The process uses the *Purchases* of the customer and the *Price Catalogue* of the goods as relevant data sources. As a result, it produces the *Bill* that gets printed and mailed to the customer.

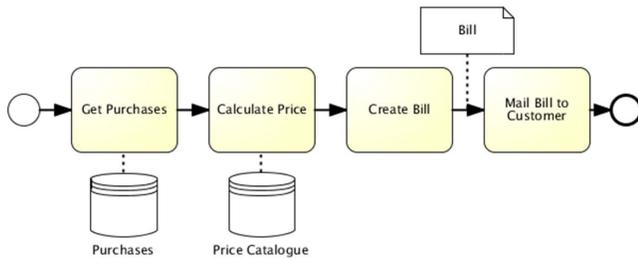


Figure 1. Billing process using a fixed pricing scheme.

Today, we see time-dependent prices (airline fees), variable pricing schemes for different customer groups or even individual customers (banking and telecommunication), as well as prices that change when purchases are bundled and that depend on sales targets (retail). Consequently, determining the price of a good requires increasingly complex computations and is influenced by various business goals and policies. In particular in the telecommunications industry, new services and service bundles are created frequently and often only exist for a certain time. Services are delivered based on policies and the processes to bill these services accurately have to be adjusted frequently. Customers should also be provided with choices how they receive and pay bills. Failure in delivering easy to understand and accurate bills leads to delayed payments, may require to adjust or correct bills, and can reduce customer satisfaction. Customer complaints about incorrect bills have to be handled with human-intensive processes, which can impact profitability. Furthermore, it becomes increasingly difficult to determine the exact costs of certain service bundles, which increases the need to test the profitability and billing schemes for new services before they are deployed, cf. [4].

*Scenario 1 — Embedded Rules:* In a Scenario 1 evolution of the billing process, additional paths are added to the process model to capture a more sophisticated pricing scheme that would, for example, take the type and age of the customer (*corporate* or *consumer/below 27*) into account. Figure 2 illustrates how the *Calculate Price* activity from Figure 1 now evolves into a separate subprocess that is

used by the billing process. Additional activities encode the different pricing (rule) calculations and rule-based decisions are embedded into the gateway logic. The data handled by the process can also be assumed to become more dynamic. Prices in the subscription rates data base are changed more frequently and different rates for different customer types must be stored. Bills are stored separately to enable access by other business processes. The customer data base becomes also more important to this process as the status and other attributes of the customer such as age now influence the billing.

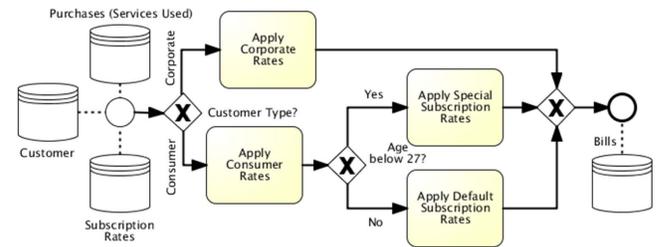


Figure 2. Scenario 1: Pricing rules encoded by a subprocess.

BPMN supports Scenario 1 very well by providing different types of gateways and the ability to capture the gateway conditions precisely. Furthermore, exceptions, error handling, and compensation can be well modeled, which we did not include as they quickly make a diagram cluttered. Despite its moderate size, the model is not so easy to understand although data annotations and other details are still missing. It becomes apparent why organizations move towards Scenario 2. SBVR does not play a significant role in this scenario as business rules are not yet made explicit. However, process models would gain precision when process modeling tools built and provided an SBVR vocabulary.<sup>1</sup>

An SBVR vocabulary can be used independently of business rules, e.g., to represent a dictionary of activities, data objects, and roles used in a process modeling tool. SBVR focuses on the *meaning* of vocabularies and rules, i.e., the semantics. In a nutshell, an SBVR vocabulary is built out of *noun concepts* representing objects and *fact types* representing relations between objects. Based on both, propositions such as for example “a purchase has a purchase date” can be expressed. Note that SBVR does not enforce a specific syntax, but is open to accommodate different notations. An example of a popular and SBVR-compliant notation is RuleSpeak [5].

We encounter a first issue with the current state of the SBVR standard. The SBVR specification relates business vocabularies to UML business object models, OWL, and other standards, but does not provide an interchange format. Importing other data representations into an SBVR

<sup>1</sup>The Signavio tool ([www.signavio.com](http://www.signavio.com)), which we used to create the BPMN models, is one of the first that allows users to build a dictionary (not yet an SBVR vocabulary) on-the-fly during modeling.

vocabulary seems to be straightforward, but to the best of our knowledge a detailed mapping of SBVR from and to other languages has not yet been described in the literature. BPMN flow objects such as data, activities, and gateways can reference other data structure formats and would benefit from an SBVR interchange format.

*Scenario 2 — Explicit Navigation Rules:* In a Scenario 2 evolution, business rules are used to represent the pricing policies. The rules can be implemented using either the rules component of a BPM suite or a combined BPM/BRMS solution. The resulting process design can be simplified to the initial set of activities from Figure 1 and is shown in Figure 3. The main difference is that *Calculate Price* is now a business rule task within the BPMN process model (note the rule task marker in the upper left corner of the activity), which delegates the computation of the price to a rules engine, i.e., explicit business rules come into the picture. Furthermore, the billing process does not need access to any pricing information, which is now encapsulated by the business rules.

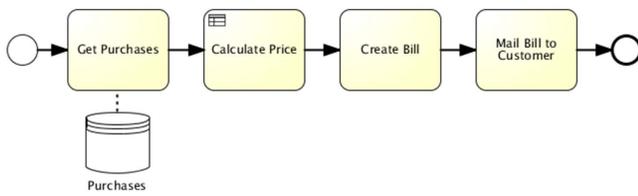


Figure 3. Scenario 2: Business rules invoked through a BPMN business rule task.

On the BPMN side nothing changes, except that process models should share the business vocabulary with the business rules. However as in the case of vocabularies, no interchange format for business rules is defined by SBVR, which is the reason why commercial and open-source rules engines provide their own languages. In terms of SBVR terminology, one would expect that all business rules are so-called structural rules. A structural rule in SBVR is a claim of necessity and describes how the business chooses to organize and structure the things it deals with. It is used to define the necessary characteristics of a noun concept, e.g., *(it is necessary that) a customer has at least one of the following: an order, an unpaid purchase, a purchase completed in the past five years*. Equivalent formulations of structural rules are: *it is necessary that p* (necessity), *it is impossible that not p* (impossibility), *it is possible that p only if q* (restricted possibility). In billing, the price calculation rules operate on business objects to compute new attribute values based on their input data, e.g., the price of a purchase (service) based on information contained in the customer and purchase (service) business objects. Even if decisions of a process are based on attribute values computed by the rules, this should not be considered as a constraint for business conduct, i.e., rules are structural, not operative, cf.

the discussion of operative business rules further below.

*Scenario 3 — Complex Navigation/Analysis:* In a Scenario 3 billing process, the number of rules used is significantly increasing and process decisions are influenced by the results returned by the rule computations. The number of process paths is growing again. Figure 4 illustrates a possible evolution of the billing process by adding a cross-selling step. One set of business rules is used to calculate the price of the purchase. Depending on the result, another set of business rules analyzes whether a service bundle should be offered to the customer that provides extended services, but would only lead to a small increase in terms of the price.

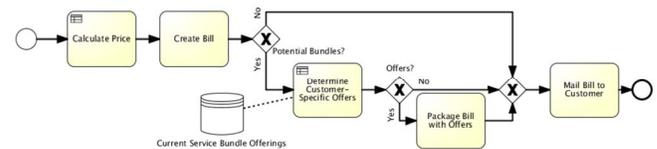


Figure 4. Scenario 3: Several business rule tasks with an increasing number of rules are used by the process.

The reduction benefits, which were achieved through the introduction of business rules in Scenario 2, begin to vanish and the process model becomes again more complicated. Additional process paths must be added during which different rule tasks are invoked. On the SBVR side, the business vocabulary and rule sets become more comprehensive. When cross-selling steps and pricing schemes become tailored to individual customers, rule sets will contain many specific rules leading to fine-grained decisions. Only a subset of rules is applicable in a specific process context, which can lead to costly computations within the rules engines that may not scale to the number of rules required. Advanced engines therefore offer some form of filtering or invocation of business rules that is guided by the process context to avoid performance problems. The consequences of changing rules may also be harder to foresee and simulation capabilities are required to predict cross-selling effects for example. Commercial rules engines offer some simulation capabilities, but there is clearly a need for further sophistication to allow designers to easily explore a wide range of scenarios, e.g., different combinations of billing and cross-selling policies.

*Scenario 4 — Rule-Guided Process Behavior:* In Scenario 4, rules are used to observe the behavior of a process and to “detect opportunities and threats in the context of the process by recognizing relevant and complex events in and around the process”. [3, p. 5] Figure 5 shows a possible evolution of the billing process towards rule-guided behavior based on the process context, e.g., the billing history of the customer and summaries and forecasts of current sales trends. The key difference in this scenario is the interaction of the billing process with other business processes that operate on a different level and take into

account very different information. In our case study, the cross-selling step for a specific customer is dynamically influenced by processes that observe the overall sales trends of specific bundles, aggregate information from sophisticated forecasting systems, and analyze the sales history of this customer compared to certain customer segments.

The process context can be captured in BPMN using so-called pools, which represent other participants (and their processes) in a collaboration. The interaction between processes is captured as BPMN message flow between pools. In Figure 5, process models have been simplified just to illustrate their main purpose and communication. Each process uses its own business rules, but may rely on aggregated rule results provided by other processes. The rules used by the forecasting process operate at a 'meta-level' compared to those used by the billing process. Price calculation rules are defined over the attributes of purchases and customers. Forecasting rules are defined over key performance indicators of sales processes and thus talk about the behavior of these processes and sales goals within a wider sales context. Although elements of a Scenario 4 process solution can be found in today's billing systems used by telecommunication providers, one will rarely find explicit process choreographies in a BPMS.

In this scenario, we see rules that are defined at the 'meta-level' encoding policies of process behavior and acting as guard rails for the process instances to steer them towards specific business goals. For example, an issued bill is combined with specific offers depending on sales targets for certain services to make the billing process contribute towards an overall sales performance goal. Business rules might rate the billing process for a specific customer based on how it performs with respect to externally set key performance indicators for sales and these rules might influence which bundles are selected during the cross-selling activity. The forecasting process with its rules thus steers the behavior of the billing process towards sales targets that have been set outside the context of the billing process itself. In contrast, only 'domain-specific' rules are used in Scenarios 1 to 3 that are defined over the *billing domain* of the billing process for an individual customer as reflected in the data provided by the customer, purchases, price catalogue, or service bundle offerings business objects.

On the BPMN side, we see inter-process communication using messages and process collaboration as a new modeling feature that we used in this scenario. Simulating and analyzing communicating processes is beyond the capabilities offered by current BPMS to the best of our knowledge. Processes begin to evolve into loosely-coupled process components to prepare the ground for Scenarios 5 to 7. On the SBVR side, subvocabularies for process goals and behavior, key performance indicators (KPI) etc. have to be maintained. Process goals and KPI should also be available in the BPMN process models. Different sets of structural business rules

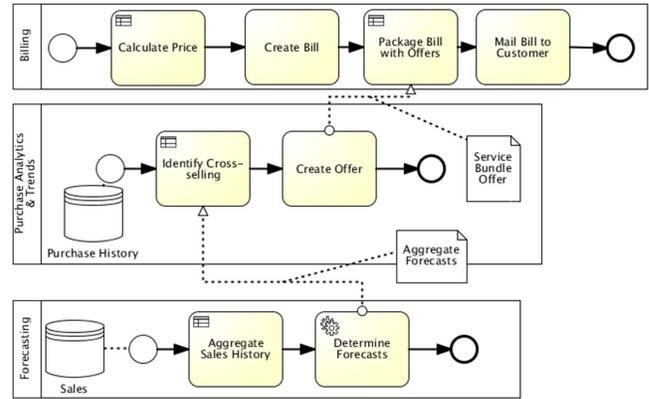


Figure 5. Scenario 4: Process Behavior guided by external processes and meta-level rules.

are maintained in the BRMS. Domain-level structural rules are specified over the process context, whereas meta-level structural rules are specified over the wider business context and cover process behavior.

*Scenarios 5 to 7 — From Rule-Driven Process Composites and Services to Fully Rule-Dynamic:* These scenarios envision the dynamic composition of processes from process and service components guided by business rules. We could not observe them in the billing domain yet. One would expect that the process and service components are built along the patterns that we described for Scenarios 1 to 4 and thus contain structural business rules. Some process components can comprise several communicating processes such as in Scenario 4, however, most process and service components will be small. The key difference for Scenarios 5 to 7 is that rules guide process composition, i.e., rules are no longer invoked by rule tasks within a process, but there are rule components/sets which take on a much more active part and now invoke the process/service components from their side. These rules are externalized from the process/service components and packaged into separate rule components that become first-class citizens (components) when designing a business solution and are no longer subordinate to processes. We expect these rules to be SBVR operative rules.

An operative business rule in SBVR formulates an understanding about the behavior of people and what form compliant behavior takes. Operative rules focus directly on the propriety of conduct where willful or uninformed actions can fall outside the boundaries of behavior deemed acceptable [2]. They formulate obligations, e.g., (*it is obligatory that*) a purchase is only opened for a registered customer. Equivalent formulations of operative rules are: *it is obligatory that p* (obligation), *it is prohibited that not p* (prohibition), *it is permitted that p if q* (restricted permission). Operative business rules specify obligations, prohibitions, and restricted permissions for process/service components and the actors involved in executing these processes and

services. They can initiate and set the guard rails for the dynamically composed process instances. Known forms of operative business rules are reaction (event-condition-action (ECA)), production, and transformation rules, see [6] for more details.

A Scenario 5/6/7 process can only be drawn “after the fact”, i.e., from monitoring its execution. This means, a BPMN diagram can be extracted by process mining the monitored process logs. However, BPMN cannot represent rules as partners in a choreography. The only way to show the rule sets and process/service components involved in a composition is by using a BPMN conversation diagram, see Figure 6. Conversation diagrams are a new model type introduced with BPMN 2.0 to draw an overview about complex collaborations where each box represents a communicating partner. However, we cannot directly distinguish different types of partners, e.g., processes and rules, and used Signavio’s coloring scheme for an initial work around (rule components depicted in grey/green).

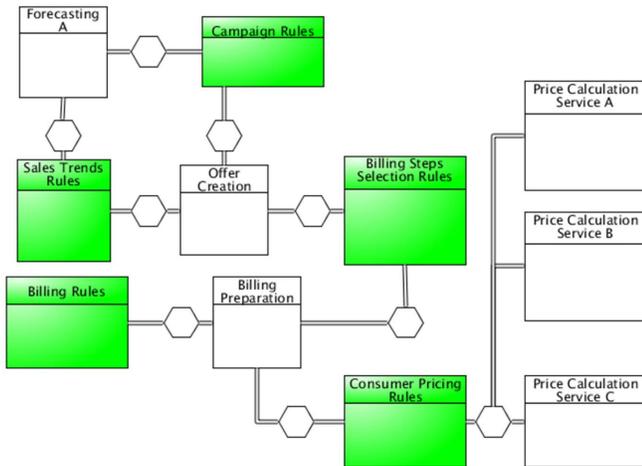


Figure 6. Scenarios 5 to 7: Rule and process/service components engaged in a conversation to dynamically create the billing process.

Each process or rule component is shown as a partner in the conversation. Between two process components, always a rule component is placed, i.e., the rule component mediates the conversation between two process components. The conversation will be started from a rule component, e.g., from the billing rules (at the left in Figure 6) that initiate the composition of the billing process. Business rules evolve from ‘structural analyzers’ of business objects to ‘direct influencers’ on the operation of the business and the behavior of actors. They decide how a process must evolve to contribute towards a larger business goal. However, in BPMN, an execution semantics for conversation diagrams is not defined, i.e., it is not yet possible to simulate such compositions of processes and rules at this level of abstraction without further refining the collaboration. For example, we cannot know from the model that only the

billing rules will initiate the conversation and that the pricing rules will each select exactly one of the pricing services instead of, for example, consulting several and recombining their results.

Compared to Scenario 5, a Scenario 6 conversation would be more fine-grained and show more participants representing specialized services as well as more specialized rule sets handling and reacting to specific business situations. Designing the right process and rule components and detecting interaction problems and unwanted behavior becomes the major challenge. Change affects rule as well as process components with the same advantages and disadvantages as discussed in Scenario 4.

On the SBVR side, the role of operative business rules must be further clarified. How can the formulated obligations, prohibitions, and restricted permissions be enforced in a dynamic business process? What extensions are required to rules engines? Furthermore, SBVR only considers the declarative representation of rule knowledge. When rules drive processes, they become (directly or indirectly) engaged in a conversation with process and service components. How can this operational and communicative aspect of business rules be addressed as it is common for example in ECA rules? At the technical level of implementation, business rules can be encapsulated as components in the Service Component Architecture (SCA) where they interact with other SCA process components by exchanging messages.

In a Scenario 7 conversation, the rule sets can evolve through a learning and change process, i.e., existing rules get revised automatically and new rules are learned. This dynamic evolution of business rules clearly lies outside the representational capabilities of BPMN and SBVR. Nevertheless, SBVR can represent rules at different stages of evolution and a BPMN diagram can be used to document the steps of the learning process and what data sources it involves.

#### IV. KEY PATTERNS AND IMPLICATIONS FOR THE STANDARDS

Summarizing the process and rule designs obtained for the seven scenarios, it seems to us that four key patterns of rule usage can be extracted, of which patterns I and II can be observed today, whereas III and IV rarely exist:

- I Embedded implicit rules (Sc. 1): Process paths are used to encode rules and decisions, no explicit rules are used.
- II Explicit rule tasks (Sc. 2/3): Business rules and business processes are distinguished and modeled using different approaches. From a modeling point of view, it does not seem relevant whether the rules component of a BPM suite or a separate BRMS is used. The key question of what goes into the rules and what goes into the processes has to be answered for both technology variants.

III Rule guidance across processes (Sc. 4): Rules are defined for different contexts and rule computation results are communicated across process boundaries to influence the behavior of other processes.

IV Active and learned rules (Sc. 5/6/7): Rules steer process components and may change during execution of the end-to-end process.

Scenarios 2 and 3 differ in the number of rules used. There is clearly a difference between using a few dozen rules and using several thousands of rules. How this difference in quantity leads to a new quality in designing business solutions requires further investigation, however, at this state of investigation it does not seem to justify a separate pattern as problems occur more on the side of maintaining and processing very large rule sets, whereas the principal pattern of how the rules are used is the same for both scenarios.

Scenario 7 definitely differs from 5 and 6, because not all rules are developed at design time, but some rules may be learned or modified during the life time of the business system. The feedback loop from the monitored results to new rules is an interesting area of research, however, the rule pattern seems to be the same to us for all three scenarios: they share the challenge to design, simulate, execute, and monitor rule-driven process compositions.

Our case study also identified various gaps for the two standards. For BPMN, Scenarios 1 to 3 do not suffer from significant standard gaps. Open challenges include best practices for embedding business rule tasks into processes (what goes where), managing several rule sets in a process, and modeling complex process flows with exceptions, compensation, and error handling. Scenario 4 would benefit from an explicit representation of process goals and KPI in BPMN that can be shared with the vocabulary used on the SBVR side to specify meta-level rules setting guard rails for process behavior. Furthermore, the simulation and execution of BPMN models containing several pools of communicating processes is currently a challenge for tool and execution providers. Scenarios 5 to 7 identify open questions around the modeling of rule-driven, dynamic process compositions, conversation diagrams, and self-learning processes. On the SBVR side, we identified more substantial gaps such as for example the need for an interchange format for vocabularies and rules, the need to ensure the consistency across large (sub)vocabularies and rule sets, support for a context-based invocation/filtering of rules, and a better understanding of structural vs. operative rules, the latter requiring the standardization of an operational semantics for rules.

Table I summarizes our findings about the number and types of rules used by the scenarios. Scenarios 2 and 3 use domain-level rules that operate within the process context, but differ in rule granularity and number of rules used. Scenario 4 introduces meta-level rules, i.e., rules that observe process behavior and goals. Scenarios 5 to 7 introduce

	1	2/3	4	5/6/7
Domain-level rules	no	some/many	many	many
Meta-level rules	no	no	some	many
Structural rules	no	yes	yes	yes
Operative rules	no	no	no	yes

Table I  
RULE PATTERNS USED BY THE SEVEN SCENARIOS.

operative rules that actively steer the end-to-end business processes, whereas structural rules are used by all scenarios except Scenario 1. Only in Scenario 7, rules change through a learning process at runtime, whereas rules are static (designed by humans) in all other scenarios.

It is also worth summarizing the scenarios from the perspective of change. Change clearly accelerates and broadens in scope from Scenario 1 to 7. In Scenario 1, the human (business) user initiates the change of a business process, which requires intensive involvement of the IT department. From Scenario 2 to 7, change can happen at the rules or process side. Due to their declarative nature and formulation in a controlled natural language, rules can often be changed directly by the business, whereas process changes usually remain a development task. Until Scenario 6, change is only initiated by a human, but starting with Scenario 4, rules contribute deeper insights that influence a human-initiated change. Only in Scenario 7, change initiation shifts from the intelligent human to the intelligent system. The componentization and packaging of processes and rules helps to structure change in Scenarios 5-7 and to manage increasingly complex change scenarios. Predicting the impact of change is a major issue and requires additional controls, see [7].

## V. RELATED WORK

Numerous publications exist that investigate the relationship between processes and rules at the level of the modeling languages, e.g., [8], [9], [10], [11], [12], [13]. Unfortunately, the different pieces and results have not been systematically put together—a task that this paper must also leave unaddressed. Process flexibility is surveyed in [14]. Comparisons of process and rule languages are presented in [15], [16]. An introduction into SBVR and a discussion of possible use cases is given in [17], which also points to the problem of *how* to enforce a rule. One possible initial answer is given in [18] that demonstrates how applications can be synthesized from SBVR rule specifications. Rules play an active part to drive process composition and adaptive business collaboration in [19], [20]. A conceptual model for inter-organizational processes that includes obligations is described in [21]. Rules replace process models completely in declarative approaches, see for example [22]. Coordination theory as introduced in [23] could serve as a starting point to better understand the role of operative business rules in rule-driven dynamic processes. The work

provides a systematic framework for the coordination of actors, tasks, and resources in business processes. Different types of dynamic business processes are discussed and related to dynamic business applications in [24]. Machine learning techniques for the acquisition of business rules, which is a prerequisite to master Scenario 7, have been an active research area for several years, see for example [25].

## VI. CONCLUSION

Starting from seven scenarios of process-rule interaction recently proposed by Gartner, we derive four key patterns of rule usage in business processes from a case study of customer billing scenarios. The patterns differ in their usage of domain- vs. meta-level rules and structural vs. operative rules. We review the BPMN and SBVR standards to the extent to which they support these patterns and identify some substantial gaps.

## REFERENCES

- [1] Object Management Group, “Business Process Model and Notation 2.0 (BPMN),” 2011, OMG document number formal/2011-01-03.
- [2] Object Management Group., “Semantics of Business Vocabulary and Business Rules 1.0 (SBVR),” 2008, OMG Available Specification, OMG document number formal/2008-01-02.
- [3] J. Sinur, “The art and science of rules vs. process flows,” Gartner, Research Report G00166408, 2009.
- [4] T. Graham and D. Baker, “The evolution of billing: From paper to e-payments,” TM Forum, Business Intelligence Quarterly April, 2011.
- [5] R. G. Ross, “The RuleSpeak business rule notation,” *Business Rules Journal*, vol. 7, no. 4, 2006.
- [6] G. Wagner, “Rule modeling and markup,” in *In Reasoning Web*, ser. LNCS, vol. 3564. Springer, 2005, pp. 251–274.
- [7] J. Hoogland, “Change in control,” in *7th Int. Conf. on Business Process Management (BPM)*, ser. LNCS, vol. 5701. Springer, 2009, pp. 28–30.
- [8] M. zur Muehlen, M. Indulska, and K. Kittel, “Towards integrated modeling of business processes and rules,” in *19th Austr. Conf. on Information Systems*. Australian Computer Society, 2008.
- [9] A. K. Ghose and G. Koliadis, “Auditing business process compliance,” in *5th Int. Conf. on Service-Oriented Computing (ICSOC)*, ser. LNCS, vol. 4749. Springer, 2007, pp. 169–180.
- [10] G. Governatori and A. Rotolo, “A conceptually rich model of business process compliance,” in *7th Asia-Pacific Conf. on Conceptual Modelling (APCCM)*. Australian Computer Society, 2010, pp. 3–12.
- [11] A. Charfi and M. Mezini, “Hybrid web service composition: business processes meet business rules,” in *2nd Int. Conf. on Service-Oriented Computing (ICSOC)*. ACM, 2004, pp. 30–38.
- [12] S. Goedertier and J. Vanthienen, “Business rules for compliant business process models,” in *9th Int. Conf. on Business Information Systems (BIS)*, ser. LNI, vol. 85. GI, 2006, pp. 558–572.
- [13] P. Bollen, “BPMN as a communication language for the process- and event-oriented perspectives in fact-oriented conceptual models,” in *OTM Workshops*, ser. LNCS, vol. 5872. Springer, 2009, pp. 639–648.
- [14] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. M. P. van der Aalst, “Process flexibility: A survey of contemporary approaches,” in *Advances in Enterprise Engineering*, ser. LNBIP, vol. 10. Springer, 2008, pp. 16–30.
- [15] M. zur Muehlen and M. Indulska, “Modeling languages for business processes and business rules: A representational analysis,” *Int. J. Information Systems*, vol. 35, no. 4, pp. 379–390, 2010.
- [16] R. Lu and S. W. Sadiq, “A survey of comparative business process modeling approaches,” in *10th Int. Conf. on Business Information Systems (BIS)*, ser. LNCS, vol. 4439. Springer, 2007, pp. 82–94.
- [17] M. H. Linehan, “SBVR use cases,” in *Int. Symp. on Rule Representation, Interchange and Reasoning on the Web (RuleML)*, ser. LNCS, vol. 5321. Springer, 2008, pp. 182–196.
- [18] A. Marinos and P. J. Krause, “An SBVR framework for RESTful web applications,” in *Int. Symp. on Rule Interchange and Applications (RuleML)*, ser. LNCS, vol. 5858. Springer, 2009, pp. 144–158.
- [19] B. Orriens and J. Yang, “A rule driven approach for developing adaptive service oriented business collaboration,” in *Int. Conf. on Services Computing (SCC)*. IEEE, 2006, pp. 182–189.
- [20] J. Koehler, “The role of BPMN in a modeling methodology for dynamic process solutions,” in *2nd Int. Workshop on BPMN*, ser. LNBIP, vol. 67. Springer, 2010, pp. 46–62.
- [21] J. Ghattas and P. Soffer, “Facilitating flexibility in interorganisational processes: a conceptual model,” *Int. J. Business Process Integration and Management*, vol. 3, no. 1, 2008.
- [22] W. M. P. van der Aalst and M. Pesic, “DecSerFlow: Towards a truly declarative service flow language,” in *The Role of Business Processes in Service Oriented Architectures*, ser. Dagstuhl Seminar Proceedings, vol. 06291, 2006.
- [23] T. W. Malone, K. Crowston, and G. E. Herman, *Organizing Business Knowledge: The MIT Process Handbook*. MIT Press, 2003.
- [24] J. R. Rymer and C. Moore, “The dynamic business applications imperative,” Forrester Research Report, 2007.
- [25] Y.-C. Huang, B. Selman, and H. A. Kautz, “Learning declarative control rules for constraint-based planning,” in *7th Int. Conf. on Machine Learning (ICML)*. Morgan Kaufmann, 2000, pp. 415–422.