

Towards a Rule Modeling Framework for Context-aware Smart Service Systems

Thi My Hang Vu^{1,2,3,*}, *Thanh Thoa Pham Thi*⁴, *Thang Le Dinh*¹

¹School of Business, Université du Québec à Trois-Rivières, Canada

²Faculty of Information Technology, University of Science, Ho Chi Minh city, Vietnam

³Vietnam National University, Ho Chi Minh city, Vietnam

⁴Faculty of Business, Technological University Dublin, Ireland

Abstract. Since business rules aim at enforcing regulations in an organization, they are critical in governing business activities from a managerial standpoint. On the other hand, another type of rules has emerged in context-aware service systems: context rules. Context rules are employed for context reasoning to recommend and operate the right services in an appropriate manner. In this sense, context rules ensure the smartness of services in smart service systems. For decades, researchers and practitioners have addressed rule modelling and rule management in information systems and business services. However, in relation to context-aware services in smart service systems, there is a lack of exploring the rule aspect, especially considering how business rules and context rules are involved in such a system. The purpose of this paper is to propose a rule modelling framework (called RuCBS framework) for expressing rules in context-aware smart service systems over the three aspects of service science (Management, Science, and Engineering). The framework presents concepts, a meta-model that connects these concepts, and rule patterns. The framework is validated with a case study on banking services. Future research directions on rules in context-aware smart service systems are also discussed.

1 Introduction

Nowadays, a smart service system is defined as a service system that can adapt to a constantly changing environment to benefit customers and providers thanks to business analytics and cognitive systems [1]. Consequently, the use of context-awareness to obtain information and insights to offer adapted services automatically is the future way of smart service systems [2]. Context-aware smart service systems are emerging to enable the context awareness and the adaptability to their changing contexts in smart service systems [3]. However, designing and implementing a context-aware smart service system is still a complex task due to the lack of an adequate model support in awareness and pervasive environment.

Rules, also often termed business rules, are related to the regulation aspect within an organization that are critical in governing business processes/activities from a managerial standpoint. In context-aware smart service systems, another type of rules has emerged:

* Corresponding author: thi.my.hang.vu@uqtr.ca or vtmhang@fit.hcmus.edu.vn

context rules, which are employed for context reasoning to recommend and operate the right services in an appropriate manner. In this sense, context rules ensure the smartness of services. For decades, rules and context-aware services have emerged in their respective domains. However, a unified framework for modelling rules over the three aspects of a service system (management, science, and engineering) [3] is still lacking.

In this paper, we propose a rule modeling framework (RuCBS) for expressing rules in context-aware smart service systems, including concepts, a meta-model that connects these concepts, and rule patterns.

This paper is organized as follows. Section 2 describes the theoretical background to present concepts used throughout this paper. Section 3 introduces the research design to conduct the study. Section 4 continues with the RuCBS framework, including concepts, models, and methods. Section 5 presents the validation of the framework to demonstrate the validity of the proposed framework. Finally, Sections 6 and 7 presents our discussions, conclusions and future research directions.

2 Theoretical background

This section clarifies key concepts related to rule modeling from the business perspective and context-aware systems, which will serve as the foundation for this paper.

2.1 Rules

Rules, according to the Oxford dictionary, are the regulations or principles governing activities or processes within a system or an organization. Business rules, a specific case of rules for the business sector, are defined as “*compact, atomic, well-formed, declarative statement*” for business regulations, policies or constraints [4, 5]. They are critical to ensure the correctness, validation, and execution of any information systems (IS). Business rules can be action (production) rules or deductive rules [6, 7]. The former performing on business objects can be determined from regulations, policies, or mandates. The latter are calculated by the application of various mining techniques.

In any rule-based system, business vocabularies and business rule modeling tools are the two most important elements. Business vocabularies determine business concepts/terms used for describing business rules, whereas rule modeling tools offer methods for business rule expression. Based on the classification proposed by [8], business vocabularies provide two kinds of vocabularies for rule modeling: nouns for presenting business concepts/terms and verbs for presenting relationships or properties of the business concepts/terms. A business rule can be described by connecting these business vocabularies in an appropriate way using rule modeling languages or tools (e.g., graphical languages, decision trees, and tables, semantic-based representation [5]).

2.2 Service and context-aware services

From the service-dominant logic perspective, a service is considered as the fundamental basic of exchange between business entities (i.e., stakeholders who have distinct goals such as consumers or providers [9]) to co-create business value [10]. In information systems, a service is implemented through business processes as the configuration of resources for the benefit of others.

Context is defined as any information, which can help to identify an entity within its environment [11]. Context information can be environmental context (i.e., context data coming from surrounding factors such as time, location, regulations), user context (e.g., user

profiles, user preferences or interests), or sensor context (i.e., context data derived from sensors) [11]. From the viewpoint of knowledge management studies, context information will be classified into the following knowledge components: know-what, know-who, know-how, know-when, and know-where [3]. In service-oriented approaches, know-what concerns services or products; know-who relates to business entities (customers, providers, or organization); know-how covers business processes or activities used in the definition or operation of a service; know-when and know-where refer to location and time factors, respectively.

More specifically, an expression of a context is “A « stakeholder » (*know-who*) performs « operations » (*know-how*) on « objects » (*know-what*) at « time » (*know-when*) in « a location » (*know-where*) because of « a contract » (*know-with*) to be consistent with « a business rule » (*know-why*)” [3].

Accordingly, context-aware services are services that could recognize and self-adapt to their context. In those services, it is essential to link and adjust business rules so that they can be appropriate to contextual information. In this sense, business rules refer to know-why knowledge that explains or governs business processes/activities [3].

3 Related work

For many years, business rule modeling has been studied in the literature; however, there is still a lack of focus on business rule modelling in context-aware service systems. This section investigates recent and relevant studies related to business rule modelling approaches and tools in general, those that consider context awareness in particular.

In general, a **business rule** can be expressed through natural languages, graphical diagrams (e.g., UML, BPEL, BPMN) [12–14], decision tables or trees [15, 16], and semantic-based representation [4, 16–18]. Semantic-based approaches, with their rich expressiveness, powerful reasoning, and ease of rule execution and transformation, appear to be the most efficient rule modelling tools. In a semantic-based solution, ontologies are primarily used for describing business vocabularies, whereas logical formulas or ontology query languages (e.g., SPARQL) are dedicated for expressing business rules. The Semantics of Business Vocabulary and Business Rules (SBVR), an OMG standard for business people, consists of metamodels and pre-defined logical formulations for describing business vocabularies and rules in a manner that facilitates rule exchanges among organizations [18]. Other studies have focused on transforming business rules into executable rules in AI-based systems, or more precisely, in rule-based systems. For example, there is a study that proposes Tier-Rules, a platform to transform and manage business rules written in natural language in executable rules to enhance collaboration between business team and IT team in the rule implementation [19]. Another study focuses on an automated transformation tool that converts semantic-based business rules into UML class diagrams to ensure the correctness and ease the execution of the business rules [20].

Many studies have examined **context rules** from the perspective of pervasive information systems (e.g., smart homes, smart cities), with less focus on business aspects of smart services. Context rules can be implemented as low-level executable rules/constraints related to databases or knowledge bases. For example, a recent study proposes an API-based framework for managing and executing context-aware rule-based services [21]. Another example is MIALinx, which concentrates on representation, transformation, and execution of rules performing on sensor context for smart bus systems [22]. In other studies [23–27], a high-level rule model for expressing context rules is being proposed. For instance, a rule-based solution for context-aware applications built upon Event-Condition-Action (ECA) patterns is suggested in [23, 24]. ABC-RuleMiner provides intelligent services through the use of association rule mining on temporal, spatial, and social context of smartphone users

[25]. A generic multi-level context model is also defined in [26] for hierarchically structuring context rules. In [27], the authors proposed an ontological context rule modeling for detecting energy waste context.

According to our observation, we are still lacking a comprehensive model that covers all business and contextual aspects of rule systems in smart services.

4 Research Design

This paper seeks to answer the following research question: *How to specify different types of rules in a context aware smart service systems for supporting context reasoning?*

The design science research (DSR) methodology [28, 29], which is a suitable and efficient method for designing and evaluating scientific artefacts, is used for the development of the rule modelling framework for context-aware smart service systems. The main stages of carrying out this DSR-based study are summarized below.

Problem Identification. The first stage is concerned with identifying the research problem. According to the findings of the related works analyzed in Section 2, there is a need for a rule modeling framework that encompasses all three levels of service science (Management, Science, and Engineering).

Objective Definition. The second stage determines research objectives to solve the identified problem, including identifying concepts and their relations for rule modeling in context-aware services, proposing methods for describing rules, and validating the proposed framework through a case study.

Design and Development. The third stage is concerned with designing and implementing artefacts for rule modelling in context-aware services. These artefacts, according to a DSR-based methodology, consist of constructs, models, and methods [28, 29]. *Constructs* relate to the main concepts used for rule expressions in context-aware services. *Models* describe relationships between constructs, which are represented by using UML notation [30]. Finally, *methods* refer to a collection of logic formula-based patterns for rule expressions.

Demonstration and evaluation. This stage is concerned with validating the proposed framework in the banking industry as a case study.

5 Rule modeling

This section introduces the paper's main contribution: a framework for rule modeling in context-aware services (hereafter called RuCBS). This framework is based on an existing framework for context-aware service modeling defined in our previous works, named Context-based Service (CBS) Framework [9, 31–33]. CBS Framework has identified fundamental concepts for smart service modeling such as *business entity*, *service proposal*, *service creation*, *service operation*, *business process*, and *business activity*. However, the framework has not clarified yet the regulation dimension of context-aware services. RuCBS is proposed to fill this gap by incorporating new *artefacts* related to rules modeling for context-aware smart service systems into the CBS framework, such as *RuCBS concepts* or *constructs*, *RuCBS models*, and *RuCBS patterns*. These artefacts will be further discussed in the next section.

5.1 RuCBS Concepts

RuCBS concept defines three rule types: Business rules (BR), Context rules (CR), and Integrity rules (IR), corresponding to *Management*, *Science*, and *Engineering* levels of service science, in that order [33]. Table 1 summarizes concepts defined in each dimension.

Table 1. RuCBS Concepts

Level	New concepts identified in this paper	Concepts reused from the CBS Framework
Management	Business rule (key concept)	Business Entity, Customer, Provider, Service Proposal, Agreement, Value Proposal
Science	Context rule (key concept), Condition, Action, Temporal Event, Explicit Event, Implicit Event	Service System, Service Creation, Service Objective, Resource, Event, Context (classified into Context-Who, Context-What, Context-How, Context-When, Context-Where, Context-Why)
Engineering	Integrity rule (key concept), Scope, Risk	Service Operation, Business Process, Business Object, Business Activity, Dynamic State

Business rules (BR) are derived from goals, policies, and contracts to ensure the regulation dimension at the *management* level. Other concepts related to BRs at this level include *business entity*, *customer*, *provider*, *service proposal*, *value proposal* and *agreement*. *Customers* and *providers* are *business entities* with distinct goals [9, 33]. A *service proposal* (e.g., Account, Credit Card) is the application of business entities' competencies, skills, or knowledge to benefit another business entities to create a *value proposal* [34]. A business rule can be defined within an *agreement* (e.g., Visa Agreement) to govern the use of service proposals.

Context rules (CR) are used to reason about context to propose and operate services in the most appropriate way at the *science* level. Each context rule specifies that an (some) *action(s)* must be taken when context *conditions* are satisfied. A context can be classified into six context types listed below: *context-who* derived from business entities (e.g., customer's profile), *context-what* concerned with what will be proposed to customers (e.g., service proposal, service operation), *context-how* related to business process/activities, *context-when* indicating temporal factor (e.g., time, season, occurrence of an event), *context-where* indicating location factor (e.g., ATM, onsite, online), and *context-why* referring to service objective. An *event* is an important element to acquire and reason a context, which is defined as something that may happen in a value co-creation network [32]. To facilitate context acquisition, RuCBS clarifies three types of events: *temporal event*, *explicit event*, and *implicit event*. Temporal events are specified by time values or time expression (e.g., end of month, or a time range from 01/01/2022 to 31/03/2022) [6]. *Explicit events* occur when a user or other system performs an explicit action or request (e.g., a user chooses the function "cash withdrawals", a third-party payment provider sends an account verification request). *Implicit events* can occur with an exception or other events (e.g., card error).

Integrity rules (IR) represent the implementation and operation dimensions of business rules within an information system in the *engineering* level. The *scope* of an IR, relating to context-what, includes a list of *business objects* (or classes in software system development) that will be affected by this IR. A *risk* is a business activity, relating to context-how, defined within a business object that can trigger this IR when the business activity is carried out. For example, a risk of the integrity rule *Credit card limitation is €3000* is Creation of a Debit transaction associated to the Transaction business object. Each time a debit transaction using a credit card is carried out (i.e. created), the related integrity rule is checked.

5.2 Models

Fig. 1 presents a meta-model of the proposed framework, which is an extension of the CBS model [28], using UML notation. The model describes relationships between the rule concepts defined in Table 1 and the concepts of a context-aware smart services over the three levels Management, Science, and Engineering.

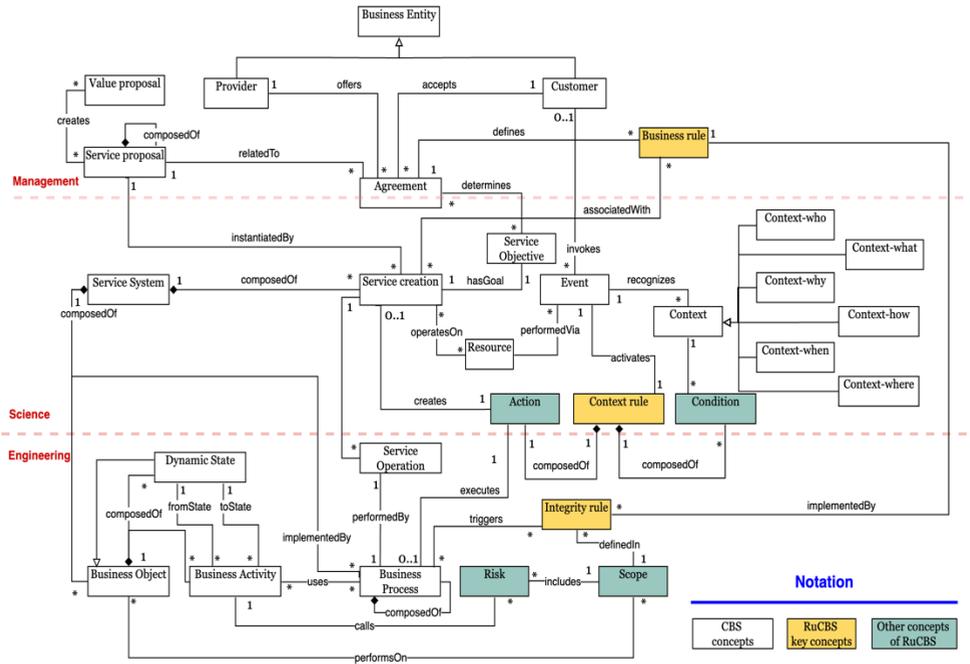


Fig. 1. RuCBS meta-model represented by UML notation

A **business rule (BR)**, as shown in Fig. 1, is *defined* within an agreement that relates to business entity (know-who) and service proposal (know-what). At the science level, a business rule is *associated with* some service creations and vice-versal to govern these service creations. At the engineering level, a business rule is *implemented* by one or several integrity rules. For example, the service creation “Open credit card” can include the business rules that monitor the cash withdrawal limits relating to the credit card, which will be implemented differently depending on the card type (e.g., golden credit cards, standard credit cards).

A **context rule (CR)** is *composed* of a set of context conditions and an action. Each condition is associated with context information (context-who, context-what, context-how, context-where, context-when). Events supports the *recognition* of context information. For example, from the event “A customer selects withdrawal function from an ATM”, context-who (customer profile), context-what (card), context-why (withdrawal), and context-where (ATM) are derived. When all context conditions are met, this CR's action is called, which then *creates* the appropriate service creation, or *executes* the relevant business process. At the **science** level, context rules are employed to reason context information to create the service creation appropriate to the recognized context. At the **engineering** level, context rules, on the other hand, are applied in reasoning context-what related to service creation, and another context information (e.g., context-when, context-where) to execute a relevant business process (context-how) to answer the question of how service is operated in a specified context.

The business process activated by context rules can *trigger* several **integrity rules (IR)** during its execution. On the other hand, an IR may be triggered by various business processes. For example, the IR used for checking balances can be used in the cash withdrawal or payment processes. Each IR is *defined* within a scope (i.e., business objects operated by this IR), and a scope can be associated with multiple IRs. For instance, the Account object can be used in integrity rules to check balances or account type. A risk is defined by a scope (through the relation “*includes*”) and a primitive business activity (through the relations “*calls*”), such as creation, deletion, and modification of a business object, which can activate the associated IR when performed [35].

5.3 Methods

This section refers to a collection of logic formula-based patterns to represent the method of the proposed framework that aims at explaining how to use RuCBS concepts to describe rules in context-aware services. Three logic-based patterns, one of the most efficient representations for rules [23], are proposed to describe Business Rule (BR), Context Rule (CR), and Integrity Rule (IR).

The **business rule pattern** consists of a rule identifier, a statement that expresses business rules in natural language, a set of business entities and service creations associated with the agreement that defines the business rules.

The **context rule pattern** is based on ECA patterns (Event, Condition, Action) [23]. The pattern includes a rule identifier, an event that activates this context rule, conditions that are context-value pairs connected through logical operators (e.g., AND, OR), and an action that is a service creation (Fig. 2) or a business process (Fig. 3) invoked when conditions are satisfied. In the scenario presented in Fig. 2, when a customer performs an event, context information is recognized and passed to the context rule to initiate appropriate service creation. Following the creation of the service, the relevant business process is invoked, which can set off integrity rules. In the scenario depicted in Fig. 3, an implicit event (e.g., card error) can be triggered during the execution of a business process, which can then invoke another context rule to select an appropriate business process to handle this implicit event.

The **integrity rule pattern** expresses constraints at the engineering level for a business rule. Each constraint has its own identifier, as well as the identifier of the business rule associated with that constraint, a list of business objects determined in its scope, a list of business activities and their associated business objects defined as its risk, and a set of IF-ELSE statements that are part of the integrity rule's implementation. This rule can be triggered when a business process employs one of the business activities identified in the integrity rule's risk.

The three rule patterns and their relationships are highlighted in Fig. 2 and Fig.3. To simplify the presentation of the patterns, we only use basic notations such as orange rectangles for patterns, white rectangles for concepts in the RuCBS meta-model, and arrows for relationships between the patterns and RuCBS concepts.

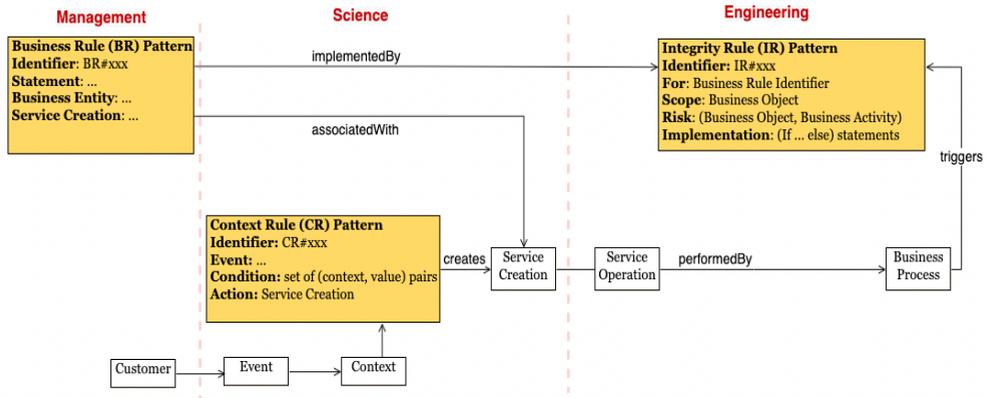


Fig. 2. RuCBS patterns' relationships for an explicit event

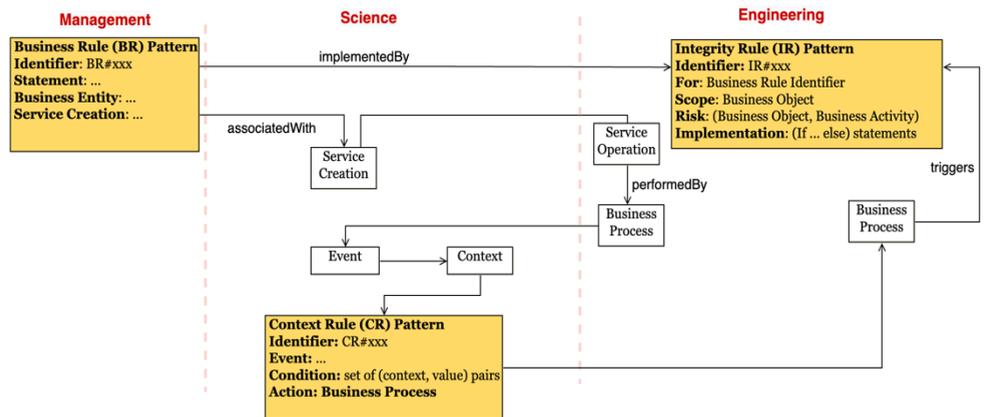


Fig. 3. RuCBS patterns' relationships for an implicit event

6 Validation

The purpose of this section is to demonstrate the validity of the proposed approach. Evaluation activities are critical in the development of any scientific artefacts based upon the design science research [36, 37]. Evaluation methods can be classified into naturalistic evaluation and artificial evaluation [38]. The former applies case studies, field studies, or surveys to validate the utility of the proposed artefacts in real-world situations. The latter deals with mathematical proofs or logical arguments to prove the reliability of the artefacts.

This paper focuses on developing a rule modeling framework that can be tailored to a specific business domain. As a result, naturalistic evaluation seems more appropriate, and a case study from the banking industry is chosen to evaluate this framework. To be more specific, the framework (concepts, models, and patterns) is applied to model three scenarios and the related business rules in relation to banking card or banking account services (labelled BR#1, BR#2, and BR#3) as examples. The business rules are derived from public information provided on banking websites.

Scenario#1

Event: A customer arrives to a bank office and asks to open a personal account for non-VIP customer.

Context information derived from the event are as follows. Context-who is Non-VIP Customer, Context-What is Standard Account, Context-Why is Opening Account, and Context-Where is Bank Office. This context rule invokes the service creation SC#11 Open a Standard Account for Non-VIP customer. This service creation relates to the following business rule:

BR#1 refers to the calculation of a monthly fee for personal standard accounts, stated as follows: “For VIP customer, the standard account management fee is free. For non-VIP customer, if customer’s monthly balance is \$3000 or more, the account management fee is waived; otherwise, he/her will be charged a fee of \$30 in the corresponding month”.

The BR#1 is associated with two service creations (SC#11-“Open a standard account for non-VIP customers” or SC#12-“Open a standard account for VIP customers”). As the event occurs and calls to perform action SC#11, this service creation is performed through different business processes based on service operations such as the fee calculation process or the card activation process. Following that, the relevant integrity rule is activated to guarantee that the business rule BR#1 is satisfied. Fig. 4 shows an example of a scenario related to BR#1 as previously described.

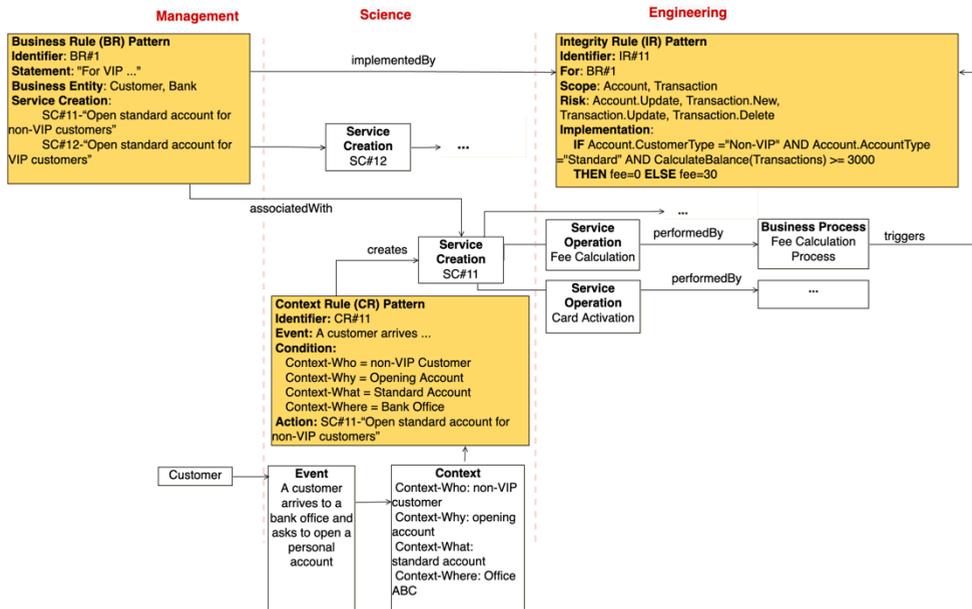


Fig. 4. Example of RuCBS patterns for describing Scenario#1

Scenario#2

Event: A student requests to open a youth account.

Context information derived from the event are as follows. Context-who is Student, Context-Why is Account Opening, Context-What is Youth Account, and Context-where is Online. Consequently, the Open youth ... account service creation is called.

The related business rule **BR#2** refers to youth account opening (SC#21) for students, stated as follows: “Students under the age of 25 can open a youth account, which includes a free monthly plan and unlimited transactions”. CR#21 is then activated. If all context conditions are met, the corresponding service creation #21 is initiated. The business process for the fee calculation can be then executed, which may invoke IR#21, as shown in Fig. 5.

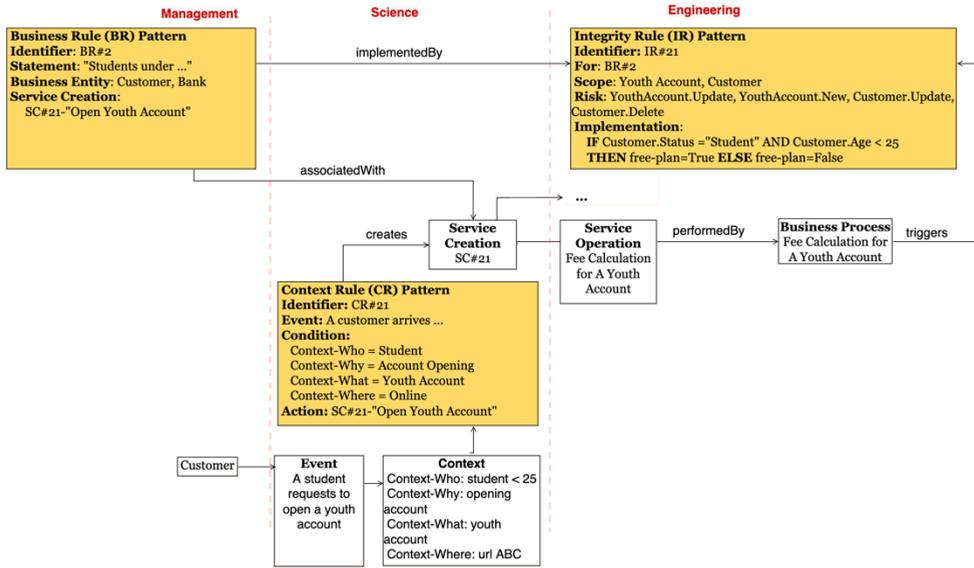


Fig. 5. Example of RuCBS patterns for describing Scenario#2

Scenario#3

Event: End of a month.

Context information derived from the event are as follows. Context-who is personal customer, Context-what is credit card, Context-why is Fee calculation. This context information is validated, and consequently, the corresponding service creation is initiated and the business process for fee calculation is executed at the end of each month (implicit event). During the business process execution, the related rule BR#3 is checked.

BR#3 refers to credit cards, stated as follows: “Interest rate applied for late payment of a credit card is 19.9%”.

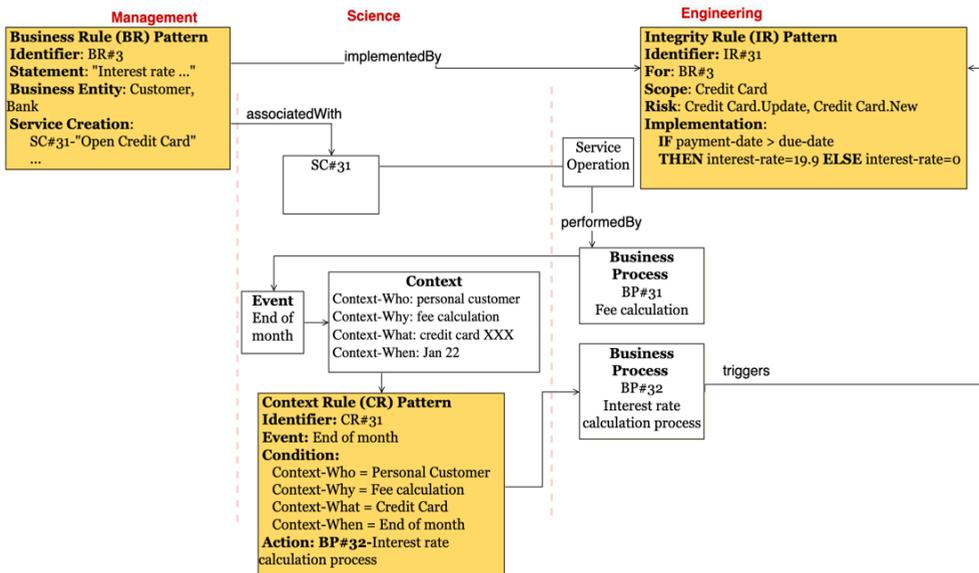


Fig. 6. Example of RuCBS patterns for describing Scenario#3

7 Discussion

In the context-aware smart service, context rules are also described with Event-Condition-Action for supporting ubiquitous computing [21]. However, an event is defined by a context, when the context is evaluated to True, then the event defined by this context may occur. In other words, “event is a context triggering the execution of a context-aware service”. Therefore, events are consistently checked over an interval of time to verify if the context defined is obtained to enable the corresponding action. For example, the room temperature is consistently checked if the room is hot (context of a room), then suitable action is called.

Similar to this approach, on the ground of information systems, a context-aware service system can be considered as an event driven system in our framework. There can be explicit events that customers explicitly make them happened, or implicit events such as temporal events- e.g., end of a month, end of a day. When an event occurs, contexts around the event are captured. Context rules define that under certain context conditions, a relevant action can be carried out. The action can be a service creation or service operation. The service executes relevant business processes. By consequence, the related business rules are validated by triggering the integrity rules which are the implementation of the business rules.

At the running time, when a context changes, but if it stills match a context rule then the corresponding service, which is defined in Action of the context rule is triggered, it can be the service creation or service operation. The service itself changes only when the business rules implemented in the service change. When the context rule changes, the process of service delivery changes, i.e., it needs to define under the new context which actions are carried out.

In the proposed framework, business rules are defined at the management level, as they relate to the service proposal and agreements between the service provider and the service consumer. The role of Context rules is to help the system to provide suitable services to users based on their context, so the context rules are described at the science level as it relates to the service creation. Meanwhile integrity rules are the implementation of business rules to constrain the business activities, so they are described at the Engineering level.

Context rules are extracted from events as shown in the rule patterns in Section 5. Similar to business rules, which are extracted from regulations, policies or agreement of the business, context rules are extracted from anticipated events that relate to all customer interactions with the service system.

The RuCBS framework has shown a complete picture of the static, dynamic, and rules aspects as well as their relationships in a context-aware smart service over the three levels of services: Management, Science, and Engineering, which have not been explored in the existing approaches.

The framework is compatible with the service-oriented architecture of information systems development. It clearly distinguishes between the front-end layer and the business logic or service layer. Events occur at the front-end layer, where user interactions with a service system happen, the events are captured and context information are evaluated to trigger suitable actions, which is actually calling suitable services at the lower level.

8 Conclusion

The paper has proposed a rule modelling framework (RuCBS) for exploring the rule aspect in smart service systems from three perspectives of service science: Management, Science, and Engineering. In terms of the research implication, this is one of the first studies that bring together three types of rules (business rule, context rule, and integrity rule) in smart service systems. Despite the fact that the importance of rule modelling is recognized by researchers

and practitioners, no research has been conducted to investigate how to describe various aspects of rules in smart service systems.

By providing a comprehensive overview of rule modeling, this study enriches the literature on the regulation dimension in smart service systems. In terms of practical implications, the proposed framework fills the gaps between engineering and business teams by clarifying the transformation and relationship between the three types of rules regarding three aspects of service science.

Furthermore, the proposed constructs, models, and rule patterns are generic enough to be reused across domains. As a result, the proposed framework can aid in the conception and development of rule management in smart service systems.

Our future work aims at using this framework in the development of a rule management system for smart services that can be validated with different domains (e.g., banking and e-commerce) to demonstrate the utility and suitability of the proposed framework.

References

1. J.C. Spohrer, H. Demirkan, *Introduction to the Smart Service Systems: Analytics, Cognition, and Innovation Minitrack*, in Proceedings of the 48th Hawaii International Conference on System Sciences. pp. 1442–1442. IEEE, HI, USA (2015)
2. T. Mo, W. Li, W. Chu, Z. Wu, *CABS3: Context-Awareness Based Smart Service System*, in Proceedings of the 2010 International Conference on Computational Intelligence and Software Engineering, pp. 1–4. IEEE, Chengdu City, China (2010)
3. T. Le Dinh, T.T. Pham Thi, C. Pham-Nguyen, L.N.H. Nam, A knowledge-based model for context-aware smart service systems, *Journal of Information and Telecommunication*, 1–22, <https://doi.org/10.1080/24751839.2021.1962105> (2021)
4. S. Goedertier, R. Haesen, J. Vanthienen, EM-BrA2CE v0.1: A Vocabulary and Execution Model for Declarative Business Process Modeling, *SSRN Journal*, <https://doi.org/10.2139/ssrn.1086027> (2007)
5. I. Graham, *Business rules management and service oriented architecture: a pattern language*, John Wiley, Chichester, England, Hoboken, NJ (2006)
6. H. Herbst, *Business Rule-Oriented Conceptual Modeling*, Physica-Verlag HD, Heidelberg (1997)
7. R.G. Ross, *Principles of the Business Rule Approach*, Addison-Wesley Professional (2003)
8. OMG, Business Rules Group - Company contributions, <https://www.omg.org/spec/company/business-rules-group/About-business-rules-group/> (2022)
9. T. Le Dinh, T.T.P. Thi, Information-Driven Framework for Collaborative Business Service Modelling, *International Journal of Service Science, Management, Engineering, and Technology* **3**, 1–18 (2012)
10. S.L. Vargo, R.F. Lusch, Service-dominant logic, *International Journal of Research in Marketing* **34**, 46–67 (2017)
11. C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context Aware Computing for The Internet of Things: A Survey, *IEEE Commun. Surv. Tutorials* **16**, 414–454, <https://doi.org/10.1109/SURV.2013.042313.00197> (2014)
12. P. Loucopoulos, W.M.N.W. Kadir, BROOD: Business Rules-driven Object-Oriented Design, *Journal of Database Management* **19**, 41–73, <https://doi.org/10.4018/jdm.2008010103> (2008)

13. G.J. Nalepa, K. Kluza, UML representation for rule-based application models with xtt2-based business rules, *Int. J. Soft. Eng. Knowl. Eng.* **22**, 485–524, <https://doi.org/10.1142/S021819401250012X> (2012).
14. F. Rosenberg, S. Dustdar, *Business rules integration in BPEL - a service-oriented approach*, in Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (CEC'05), pp. 476–479 (2005)
15. OMG, About the Decision Model and Notation Specification Version 1.4 beta, <https://www.omg.org/spec/DMN/> (2022)
16. Oracle, Fusion Middleware Designing Business Rules with Oracle Business Process Management, <https://docs.oracle.com/middleware/1213/bpm/rules-user/> (2022)
17. G.F. Nalepa, *Modeling with Rules Using Semantic Knowledge Engineering*. Springer International Publishing, Cham (2018)
18. OMG, *Semantics of Business Vocabulary and Business Rules*, v1.4, <https://www.omg.org/spec/SBVR/1.4/PDF> (2022)
19. P. Jiménez, R. Corchuelo, On the design of an advanced business rule engine. *Software: Practice and Experience*, <https://doi.org/10.1002/spe.3115> (2022)
20. M. Bonais, K. Nguyen, E. Pardede, W. Rahayu, Automated generation of structural design models from SBVR specification, *AO* **11**, 51–87, <https://doi.org/10.3233/AO-160162> (2016).
21. H. Takatsuka, S. Saiki, S. Matsumoto, M. Nakamura, *Design and Implementation of Rule-Based Framework for Context-Aware Services with Web Services*, in Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services. pp. 233–242. ACM, Hanoi Viet Nam (2014)
22. M. Wieland, F., Steimle, B., Mitschang, D. Lucke, P. Einberger, D. Schel, M. Luckert, T. Bauernhansl, *Rule-based integration of smart services using the manufacturing service business*, in Proceedings of the 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI), pp. 1–8 (2017)
23. L. Daniele, P. Dockhorn Costa, L. Ferreira Pires, Towards a Rule-Based Approach for Context-Aware Applications, in: Pras, A. and van Sinderen, M. (eds.) *Dependable and Adaptable Networks and Services*. pp. 33–43. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
24. P.D. Costa, J.P.A. Almeida, L.F. Pires, M. van Sinderen, Evaluation of a Rule-Based Approach for Context-Aware Services, in Proceedings of the IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference, pp. 1–5 (2008)
25. I.H. Sarker, A.S.M. Kayes, ABC-RuleMiner: User behavioral rule-based machine learning method for context-aware intelligent services, *Journal of Network and Computer Applications* **168**, 102762 (2020).
26. F. Burgstaller, B. Neumayr, C.G. Schuetz, M. Schrefl, *Modification Operations for Context-Aware Business Rule Management*, in Proceedings of the 2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC), pp. 194–203 (2017)
27. J. Han, Y.-K. Jeong, I. Lee, *A Rule-Based Ontology Reasoning System for Context-Aware Building Energy Management*, in Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive

- Intelligence and Computing. pp. 2134–2142. IEEE, LIVERPOOL, United Kingdom (2015)
28. A.R. Hevner, S.T. March, J. Park, S. Ram, Design Science in Information Systems Research, *MIS Quarterly* **28**, 75–105, <https://doi.org/10.2307/25148625> (2004).
 29. K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A Design Science Research Methodology for Information Systems Research, *Journal of Management Information Systems* **24**, 45–77, <https://doi.org/10.2753/MIS0742-1222240302> (2007)
 30. UML: About the Unified Modeling Language Specification Version 2.5.1, <https://www.omg.org/spec/UML/2.5.1/About-UML/>
 31. T. Le Dinh, T.T.P. Thi, N.A.K. Dam, W. Menvielle, *A Service Science Perspective on Resilience of Service Organisations*, in Proceedings of the ITM Web Conf. 38, 02002. <https://doi.org/10.1051/itmconf/20213802002> (2021)
 32. T. Le Dinh, T.T.P. Thi, Collaborative Business Service Modelling in Knowledge-Intensive Enterprises, *International Journal of Innovation in the Digital Economy* **7**, 1–22, <https://doi.org/10.4018/IJIDE.2016100101> (2016)
 33. T. Le Dinh, T.M.H. Vu, N.A.K. Dam, J. Ralyte, Smart service modeling, technical report (2022)
 34. S.L. Vargo, P.P. Maglio, M.A. Akaka, On value and value co-creation: A service systems and service logic perspective, *European Management Journal* **26**, 145–152, <https://doi.org/10.1016/j.emj.2008.04.003> (2008)
 35. T. Le Dinh, G. Fillion, Acquiring Domain Knowledge of Information Systems: The Information System Upon Information Systems Approach, *Academy of Information & Management Sciences Journal* **10**, 57–77 (2007)
 36. R. Coetsee, Towards Designing an Artefact Evaluation Strategy for Human Factors Engineering: A Lean Implementation Model Case Study, *South African Journal of Industrial Engineering* **30**, 289–303, <https://doi.org/10.7166/30-3-2244> (2019).
 37. J. Venable, J. Pries-Heje, R. Baskerville, FEDS: a Framework for Evaluation in Design Science Research, *European Journal of Information Systems* **25**, 77–89, <https://doi.org/10.1057/ejis.2014.36> (2016).
 38. J. Venable, J. Pries-Heje, R. Baskerville, A Comprehensive Framework for Evaluation in Design Science Research, in: Peffers, K., Rothenberger, M., and Kuechler, B. (eds.) *Design Science Research in Information Systems, Advances in Theory and Practice*, pp. 423–438. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)