Validation and Verification of Business Rules

Nor Najihah Zainal Abidin[2,3][0000−0003−0608−6741], Nurulhuda A.Manaf[1][0000−0002−5392−4923], Nur Amalina Jamaludin[3][0000−0003−0608−6741], and Sotiris Moschoyiannis[4][0000−0002−0164−8322]

[1] National Defence University of Malaysia (NDUM), 53000 Kuala Lumpur, Malaysia
3201334@alfateh.upnm.edu.my
[2] National Defence University of Malaysia (NDUM), 53000 Kuala Lumpur, Malaysia
nurulhuda@upnm.edu.my
[3] National Defence University of Malaysia (NDUM), 53000 Kuala Lumpur, Malaysia
amalinajamaludin@upnm.edu.my
[4] Department of Computer Science University of Surrey Guildford, United Kingdom
s.moschoyiannis@surrey.ac.uk

Abstract. The Semantics of Business Vocabulary and Rules (SBVR) OMG standard and its supplementary Date-Time Vocabulary (DTV) have been proposed for specifying business models. The rules in such models are prescribed in Structured English (SBVR-SE) making them easier to understand and removing the need, or reducing the gap, for layers of business analysts between the stakeholder and the end programmer. In this paper, we validate the SBVR rules by translating the rules into regular expressions (regex) which allows for a representation of rules by Communication Finite State Machines (CFSMs). This formal representation is then compared with the textual representation used in the global graph for validation and model verification purposes.

Keywords: Service oriented computing · Service choreography · SBVR · Behavioural modelling · Complex systems · Model transformation · CFSMs · Regular expression · Validation

1 Introduction

There are multiple well-known models for specifying a business model. A business process model written in Business Process Model and Notation (BPMN) [26] can be given in an easy-to-understand graphical notation. The characterisation of the model is similar to the Unified Modeling Language (UML) [29]. It is a well-known standard specification language which is flexible enough to be applied to a wide range of sectors, such as business, transport, health, and so on.

Semantic of Business Vocabulary and Business Rules (SBVR) [28] is an Object Management Group (OMG) standard used by business people, or the stakeholder, to express business requirements as declarative rules. SBVR is written in natural language and enables users to validate the resulting specification (terms, fact types, rules) directly as well as transform it to formal logic for automated verification purposes.

Recent works [4,16,2,18] advocate SBVR for specifying business models. In previous work, we have applied SBVR and its supplement, the *Date-Time Vocabulary* (DTV) [30], for specifying service choreographies [2,18]. Further, the *SBVR2Alloy* compilation tool [17] has been built that can automatically generate the service choreography from an SBVR model. An Alloy model [2] describes a set of constraints and performs automated analysis on the model. It produces an instance structure of model that satisfies the ordering of constraints in the input service choreography.

The service choreography approach [36] coordinates the collaboration of distributed systems across autonomous participant services [32]. Choreography focuses mainly on prescribing the ordering of the message exchange between services, according to agreed global constraints. It is key to realising value added service chains in ecosystem oriented architectures [21].

The main contribution of this paper is to validate the specification of SBVR rules for coordinating the choreography model. The validation is based on equating the expressing and the interpreting of the meaning between the regular expressions [33] with the representation of a communicating finite-state machine (CFSM) [6] and the textual representation [10].

This paper is organised as follows. Section 2 outlines the representation of an SBVR model, including business rules, in regex and CFSMs. Section 3 describes the textual representation of the SBVR model. Section 4 presents the validation of SBVR rules. Section 5 discusses related work and Section 6 includes conclusions and future work.

2 Representation of SBVR model in Regex and CFSMs

SBVR is utilised by business people since it is given in natural language to express business objects as rules, such as development of business model according to Object Management Group (OMG) [28] standard. The semantics of SBVR is represented in formal logic so it can be machine-processed allowing and assisting business experts in generating, identifying, validating, as well as administering business rules [?]. Recent works [18,17,2,25] advocate OMG standard SBVR [28] for characterising the global behaviours constraints in modelling choreography. SBVR structured English (SE) is applied for designing the rules prescribed in the multi-party conversation, and capture the ordering of global constraints in the complex interactions involved. In this section, we give a brief introduction to the SBVR OMG standard as well as an overview of the use of SBVR rules for service choreography. We then proceed with a description of how the SBVR rules are translated into regex and CFSMs.

2.1 Defining Constraints in the SBVR

The OMG standard SBVR is concisely mentioned in this section. Then, a structure of SBVR rules for specifying the choreographies are constructed.

An overview of OMG standard SBVR SBVR is a meta-language that represents the semantics of business vocabularies as well as business rules. The rules consist of Terms combine to create Fact Types, which result in a set of rules. Figure 1 shows an example of SBVR rule from [28]. The rule is constructed with the combination of fact types, modality (obligations), and quantification (each, exactly one). Rulemotion web-based SBVR editor [20] is used to create the rule.

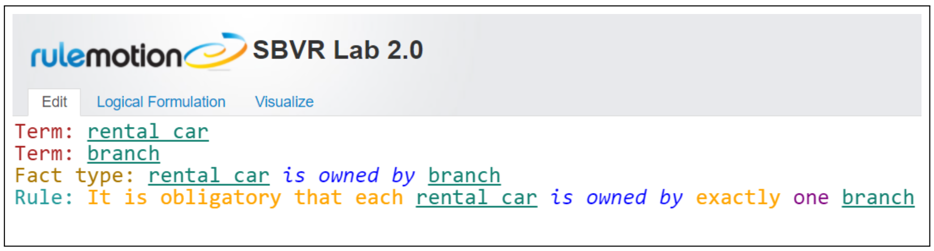


Fig. 1: Part of an SBVR model

SBVR rules for designing Service Choreographies SBVR standard has explicitly defined formatting notation [28] and the semantics for specifying SBVR rules for choreographies. As mentioned previously, **Terms** are a basis for creating Fact Types. Terms are applied to represent *Participants* who participate in the service interactions, e.g. Term: Customer, Term: Bank; *Events* as the messages exchange between participants, e.g. Term: payment; *Static constraints* characterising the domain specific constraints, e.g. Term: name; and *Time interval* illustrating the ordering of time interval associating with the same event by different participants, e.g. Term: T1, Fact Type: customer *makes* payment *at* T1, Fact Type: bank *receives* payment *at* T2. It illustrates the event, payment is made by a customer initiates the interaction, it is then followed by the same event, payment is received by a bank.

In modelling choreographies, several types of **Fact Types** has been grouped accordingly. The Set definition in [28] : set *includes* thing is used to specify both of the participant set and the event set along with their nesting group, e.g. Fact Type: accommodation *includes* apartment, Fact Type: accommodation *includes* hotel; term *verbs* term is applied to prescribe the messages exchange involved, e.g. Fact Type: bank *receives* payment. Note that *verb* can be any verbs to illustrate the sending or receiving of the messages by the participants; and the Date-Time Vocabulary (DTV) [30] is supplemented for SBVR specification. A notion of *immediately precedes* is expressed the ordering of interactions. For instance, event 1 *immediately precedes* event 2 which specifies there is no event that happens after the event 1 and before the event 2.

Subsequently, each **Rule** for designing choreography is constructed with a combination of fact types, modality (obligation), quantification, and the logical operator includes AND specifying the messages exchange are performed concurrently; OR expressing at least one of the events is selected in the messages exchange; and XOR prescribing the explicit choices of the events for executing the messages exchanged. In this paper, we focus on the obligation rules which are expressive enough for the purpose of designing choreographies.

2.2 Translating the SBVR rules into the Regex and CFSMs

This section highlights on how the SBVR rules modelling choreography is translated into regular expression. It allows the representation of rules by CFSMs to validate the consistency of the SBVR rules. This SBVR rules are specified according to the structure of rules discussed in the previous section.

Regular Expression (Regex) Regular expressions (regex) are a widely known and powerful way to manipulate text automatically. By using a regex generator [3], programmers can describe whether a given string corresponds or not to a preferred set of strings. This is beneficial for data validation, data scraping, syntax highlighting and it acts as a means for development in theoretical computer science and formal language theory [8]. Regex is a type of algebraic expression that can be used to describe languages. It is a term made up of letters (the set of inputs, Σ), numbers and the operator symbols that denote the logical operations such as exclusive choice (XOR), conjunction (AND) and inclusive choice (OR). To align the regex with the SBVR rules purposes in designing choreography, $|$ uses to illustrates parallel interaction, while $+$ to represents branching interaction and $\cdot$ depicts sequential interaction. Furthermore, the arrow ($\rightarrow$) illustrates the outgoing transition from the participant(s).

CFSMs Communication finite state machines (CFSMs) is a well-known and commonly used model. CFMSs and the regex is also useful for validating purpose [3][15]. CFSMs consists of states that ranged of $q_0, q_1, ...$ with initial state denotes as (➤) and final state denotes as (◎) . In addition, the arrow ($\rightarrow$) indicates the transition in the form of strings (*a,b,abab,...*). In this paper, CFSMs is adopted from [6] to describe a model of global specifications. To align with SBVR rules for modelling choreography, states are defined in the ranged of *0,1,2,...* and the set of inputs (Σ). Each state represents the interaction between the autonomous participants. The arrow ($\rightarrow$) indicates the transition that occurs in the choreography. The transition means a sequential or the ordering of interactions between the participants. Definition 1 presents some preparatory notations for constructing the CFSMs corresponding to the SBVR rules for modelling choreography.

Definition 1. *The following sets and notations are used to develop CFSMs for describing choreography model from SBVR model. Let p represents all participants that involved in the services interaction (service choreography) (ranged over by* $p_1, p_2, p_3...p_n$*). Given the finite states of state Q, is an automaton* $M = (Q, q_0, \Sigma, \delta, F)$ *satisfies the following condition:*

- The set of inputs is $\Sigma \equiv$ p $\rightarrow$ p':e.
- Q = set of finite states of interaction, ranged by *0, 1, 2, . . . n*.
- e = set of events, ranged by $e_1, e_2, e_3...e_n$.
- $q_0 \in Q$ is the initial state.
- $\delta \subseteq Q \times \Sigma \times Q$ is the set of transition.
- $F \subseteq Q$ is the set of final state.
- ϵ = empty string.

Translating Rules into the Regex and CFSMs Section 2.1 illustrates several types of SBVR rules capturing the specification of the complex interactions and the ordering of the interactions between participants.

The following rules are several rules in SBVR for modelling choreography. Rule A is a basis rule to specify the messages exchanged between participants. *verb* can be any verbs describing the sending (receiving) of the event. Rule B, Rule C, and Rule D prescribe the rules to illustrate the concurrent interaction (Rule B) and the alternative interaction (Rule C and Rule D). Rule C emphasises on at least one of the events is selected during the interaction while Rule D stresses on exactly one of the events in the choices must be selected by the participant. T for each rule represents the time interval where T refer to T1, T2,..,Tn. Rule E is an example of rule showing the ordering of the interactions involving the messages exchanged of event 1 and event 2.

- Rule A: It is obligatory that the participant 1 *verb* exactly one event 1 at exactly one T.
- Rule B: It is obligatory that the participant 1 *verb* exactly one event 1 and exactly one event 2 and ... and exactly one event N, at exactly one T.
- Rule C: It is obligatory that the participant 1 *verb* exactly one event 1 or exactly one event 2 or ... or exactly one event N, at exactly one T.
- Rule D: It is obligatory that the participant 1 *verb* exactly one event 1 that includes exactly one event a or exactly one event b but not both at exactly one T.
- Rule E: It is obligatory that exactly one event 1 immediately precedes exactly one event 2.

Table 1: Translation of SBVR rules into Regex and CFSMs

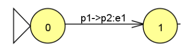
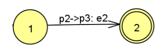
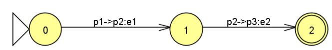
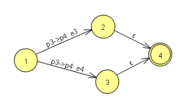
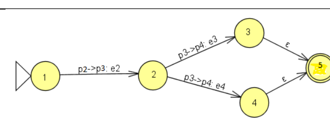
SBVR rule	Regular expression	CFSMs
Interaction 1		
Rule 1: It is obligatory that the participant 1 sends exactly one event 1 at exactly one T1 Rule 2: It is obligatory that the participant 2 receives exactly one event 1 at exactly one T2	$0 : \epsilon$ $1 : 0(\mathrm{p}_1 \rightarrow p_2 : e_1)$ $1 = \epsilon(p_1 \rightarrow p_2 : e_1)$	
Interaction 2		
Rule 3: It is obligatory that the participant 2 sends exactly one event 2 at exactly one T1 Rule 4: It is obligatory that the participant 3 receives exactly one event 2 at exactly one T2	$2 : 1(\mathrm{p}_2 \rightarrow p_3 : e_2)$ $2 = 1(\mathrm{p}_2 \rightarrow p_3 : e_2)$	
Rule 5: It is obligatory that exactly one event1 immediately precedes exactly one event2	$0 : \epsilon$ $1 : 0(\mathrm{p}_1 \rightarrow p_2 : e_1)$ $2 : 1(\mathrm{p}_2 \rightarrow p_3 : e_2)$ $2 = 1(\mathrm{p}_2 \rightarrow p_3 : e_2)$ $= 0(\mathrm{p}_1 \rightarrow p_2 : e_1)\cdot\ (\mathrm{p}_2 \rightarrow p_3 : e_2)$ $= \epsilon(p_1 \rightarrow p_2 : e_1)\cdot\ (\mathrm{p}_2 \rightarrow p_3 : e_2)$	
Interaction 3		
Rule 6: It is obligatory that the participant3 sends exactly one event3 and exactly one event4, at exactly one T1 Rule 7: It is obligatory that the participant4 receives exactly one event3 and exactly one event4, at exactly one T2	$3 : 2(\mathrm{p}_3 \rightarrow p_4 : e_3)$ $4 : 2(\mathrm{p}_3 \rightarrow p_4 : e_4)$ $5 : 3(\epsilon) \mid 4(\epsilon)$ $5 = 3(\epsilon) \mid 4(\epsilon)$ $= 2(p_3 \rightarrow \mathrm{p}_4 : e_3) \mid 2(\mathrm{p}_3 \rightarrow \mathrm{p}_4 : e_4)$ $= 2\ [(p_3 \rightarrow \mathrm{p}_4 : e_3) \mid (\mathrm{p}_3 \rightarrow \mathrm{p}_4 : e_4)]$	
Rule 8: It is obligatory that exactly one event2 immediately precedes exactly one event3 and event4	$0 : (\epsilon)$ $1 : 0(\mathrm{p}_1 \rightarrow p_2 : e_1)$ $2 : 1(\mathrm{p}_1 \rightarrow p_2 : e_2)$ $3 : 2(\mathrm{p}_3 \rightarrow p_4 : e_3)$ $4 : 2(\mathrm{p}_3 \rightarrow p_4 : e_4)$ $5 : 3(\epsilon) \mid 4(\epsilon)$ $5 = 3(\epsilon) \mid 4(\epsilon)$ $= 2(\epsilon)(p_3 \rightarrow p_4 : e_3) \mid 2(\epsilon)(p_3 \rightarrow p_4 : e_4)$ $= (\epsilon)(p_1 \rightarrow p_2 : e_1) \cdot (p_1 \rightarrow p_2 : e_2) \cdot [(p_3 \rightarrow p_4 : e_3) \mid (p_3 \rightarrow p_4 : e_4)]$	

Table 1 shows an example for characterising the interactions in choreography by applying SBVR rules. These rules are translated into the regex with the visualisation of CFSMs.

The equations in the regex represent each interaction which describes states in CFSMs. In order to have the required regular expression for the given automata, the equation from all interactions (states) must be substituted into the equation of the final interaction. All the equations of the regex need to be simplified by using the substitution method.

Interaction 1 in Table 1 depicts the occurring of the messages exchanged between the participant 1 who sends the event 1 (Rule 1 - showing it occurs at time interval, T1) and the participant 2 who receives the event 1 (Rule 2 - showing it occurs at time interval, T2). To illustrate the sending and the receiving, verb: sends and verb: receives are used. In the representation of CFSMs, there are states illustrate each interaction. For the interaction 1, CFSMs consists of states that ranged by *0* as an initial state and *1* with outgoing transition for the occurrence of the interaction 1 between participant 1 (p_1) (the sender of the event 1) and participant 2 (p_2) (the receiver of the event 1) for the messages exchanged of the event 1 (e_1), ($p_1 \to p_2 : e_1$) from state *0*. Since no interaction happens before at state *0*, it is declared as an empty string, ϵ in the regex. The numbering 1 in the regex denotes the whole interaction. Hence, there are 0 (represents the previous empty interaction) and the new interaction between p_1 and p_2, substitute in 1.

The second interaction is represented through Rule 3 and Rule 4. Both rules declare the messages exchanged of the event 2 between the participant 2 and the participant 3. According to Rule 5, this second interaction is occurred right after the first interaction previously (it is interrelated). This is a reason state *2* has incoming transition from state *1* (the previous interaction) with the input of ($p_2 \to p_3 : e_2$), as shown in the CFSMs for Rule 5. Similarly, numbering 2 is substituted by the first interaction denoted as 1 describes $\epsilon(p_1 \to p_2 : e_1)$, then is followed by the next interaction using a symbol, "$\cdot$".

Interaction 3 prescribes in Rule 6 and Rule 7. The parallel interaction declared by the logical operator **and** describes both events, the event 3 and the event 4 are sent concurrently by the participant 3, and then both events are concurrently received by the participant 4 afterwards. The last Rule 8 emphasises on the precedence of the occurrence between the interaction 2 and the interaction 3.

The regex declaration, 5 (the last equation of the regex for Rule 8) shows the whole interaction (interaction 1, 2, and 3) as defined in SBVR rules. $3(\epsilon) \mid 4(\epsilon)$ in 5 represents the parallel occurrence of the messages exchanged of the event 3 and the event 4, between the participant 3 and the participant 4. This is illustrated in CFSMs by using the fork from node 2.

3 Textual representation for SBVR model

Textual representation is adapted from [10]. The global view of a choreography (G-choreography) represents multiple interactions among the autonomous participants. The textual notations are used to represent the interaction. G::== (o) means no interaction so this interaction can be omitted; the instance: $p_1 \rightarrow p_2$: e represents a single interaction for specifying the message exchanged of the event, e between p_1 and p_2; [G;G'] is the sequential interaction between [G] and [G'], the notation ; captures the ordering of the interaction; [G | G'] defines the parallel interaction [G] and [G']; sel $\{G_1 + ... + G_n\}$ shows the branching of the possibility to choose either of the G-choreographies($G_1 + ... + G_n$).

Table 2 shows the textual representation according to the SBVR rules defined in table 1.

A single interaction notation is used to illustrate the interaction 1: Rule 1 and Rule 2 as well as the interaction 2: Rule 3 and Rule 4. Rule 5 and Rule 8 is represented by applying the sequential notation (";" means "precedes") to show the ordering of the interaction 1 and the interaction 2, and the interaction 2 and the interaction 3, respectively. Moreover, Rule 6 and Rule 7 depicts the messages exchanged of the events occur concurrently (AND) where "|" illustrates the parallel interaction.

Table 2: Developing textual representation for SBVR model

SBVR rule	Textual representation
Rule 1 and Rule 2	$(p_1\rightarrow \mathrm{p}_2 : e_1)$
Rule 3 and Rule 4	$(p_2\rightarrow \mathrm{p}_3 : e_2)$
Rule 5	$(p_1\rightarrow \mathrm{p}_2 : e_1);(\mathrm{p}_2\rightarrow \mathrm{p}_3 : e_2)$
Rule 6 and Rule 7	$(p_3\rightarrow \mathrm{p}_4 : e_3 \mid \mathrm{p}_3\rightarrow \mathrm{p}_4 : e_4)$
Rule 8	$(p_2\rightarrow \mathrm{p}_3 : e_2)$;$[(p_3\rightarrow \mathrm{p}_4 : e_3) \mid (p_3\rightarrow \mathrm{p}_4 : e_4)]$

4 Validation of SBVR rules in SBVR model

The mechanism of validating the SBVR rules for the SBVR model is by verifying (equating) the equations representing and interpreting the meaning of the regular expression and the textual representation from the corresponding SBVR model. It can be seen in Table 1 and Table 2.

In the regular expression, all interactions 1, 2, and 3 are defined as 5 : $(\epsilon)(p_1 \rightarrow p_2 : e_1) \cdot (p_1 \rightarrow p_2 : e_2) \cdot [(p_3 \rightarrow p_4 : e_3) \mid (p_3 \rightarrow p_4 : e_4)]$ (refer Table 1). This expression is obtained after the substitution of all interactions involved. It shows the sequence of the interactions specified using the SBVR rules of the illustrative of choreography model.

The textual representation specifies the ordering of all interactions involved for the corresponding SBVR rules by defining ($p_1 \rightarrow p_2 : e_1$) ; ($p_2 \rightarrow p_3 : e_2$) ;[($p_3 \rightarrow p_4 : e_3$) | ($p_3 \rightarrow p_4 : e_4$)] (refer Table 2).

Expressions interpret the same meaning of the interactions in the choreography model with the similar representing of the expressions. This verifies the specification of SBVR rules in the SBVR model.

5 Related work

[1] provides a first-order deontic-alethic logic (FODAL) to express business constraints defined in SBVR and perform a consistency check on the rule set, including alethic and deontic rules. On the other hand, we perform a validation to verify the specification of SBVR rules used to model choreography, especially on the ordering of the service interactions and the complex interaction which are used the logical operator in the SBVR rules.

We apply the regular expression (regex) and the textual repsentation for the validation process. The textual representation is adapted from [10] is used to represent Global graph of the choreography [12]. Regex proves that it is useful for verifying input of an expected pattern or structure.

Web Services Choreography Description Language (WS-CDL) is another language for choreography specification [19]. According to [19], proposes a metamodel-driven transformation technique that refines Web Service Choreography Description Language (WS-CDL) choreographies into executable Business Process Execution Language for Web Services (WS-BPEL) orchestrations using a set of Atlas Transformation Language (ATL) rules. Metrics are empirically validated using a case study of the WS-CDL process to establish their applicability, according to a paper published in [14]. Furthermore, [34] mentioned that they propose the WS-CDL language in order to maintain the features of certain ubiquitous devices. They developed and implemented a ubiquitous device coordination structure based on the WS-CDL specification. However, WS-CDL is unable to recognise and develop a method for verifying conformance to choreography specifications. [22].

Furthermore, the Decision Model and Notation (DMN) is a designing language and basic notation for representing decision rules, as specified by the OMG [27]. It is another well-known standard specification language for modelling service interactions as well as displaying them in easy-to-understand graphical notations [5][7][11]. It provides an integrated notation for decision management in the same way that BPMN does for business processes. However, DMN suggests a long technical noun phrase for each intermediate stage, whereas SBVR keeps considerably closer to what people actually say in the business world. Hence, SBVR employs more natural business language than DMN.

Instead of requiring the user to describe how to achieve an answer, the declarative approach allows users to define the constraints, actions, and outcomes of each action [31]. A declarative method based on the query/view/transformation-relations (QVT-R) standard has been used to transform Systems Modeling Language (SysML) models, according to [13]. In addition, [9] proposed two modelling patterns that describe the concepts of modelling application deployment and provide a better knowledge of declarative and imperative modelling approaches. Using a declarative approach, a previous study from [31] provided an innovative teaching framework that organises pupils and plots a course schedule with the goal of supporting the student in finishing all course subjects. A hybrid approach was used to convert declarative choreography models to imperative choreography models [37]. DecSerFlow [23][35] is also another declarative approach that was utilised as the graphical specification of service flows specifying through a set of policies rather than business rules for service choreographies. Despite the goals of our approaches are similar, that SBVR as well as its structured English (SBVR-SE) is an OMG standard that can be comprehended by humans as well as machines whereas DecSerFlow is indeed a proprietary graphical modelling language.

6 Conclusion and future work

The SBVR model, which is a declarative approach, is used an OMG standard, SBVR rules in conjunction with Date-Time Vocabulary for coordinating a choreography model.

The validation of SBVR rules has been performed in this paper by translating the SBVR rules into the regex which allows a representation of the rules by CFSMs. This formal representation is then compared with the textual representation used in the global graph.

In order to enable the users to participate in the development of the SBVR model on their own and and then transform the SBVR model into the Alloy model automatically, the SBVR2Alloy tool [17] has been developed. It can be used to express complex rules, with a focus on capturing constraints on the orderings of service interactions, including concurrent interactions [24]. The tool can be extended to include less common features of SBVR and indeed this is part of the future work planned. The ultimate goal is to develop an automated tool for modelling business rules as well as executing the corresponding SBVR model and offering a preview of all possible executions to both modellers and end-users so that the business model can be adapted or extended to better match the business need.

Acknowledgements This research is funded by Malaysian Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS)/1/2018/ICT01/UPNM/03/1).

References

1. Logic-based reasoning support for sbvr. Fundamenta Informaticae **124**(4) (2013)
2. A.Manaf, N., Moschoyiannis, S.: Generating choreographies from sbvr models. In: AIP Conference Proceedings. vol. 2184, p. 060062. AIP Publishing LLC (2019)

3. Arcaini, P., Gargantini, A., Riccobene, E.: Mutrex: A mutation-based generator of fault detecting strings for regular expressions. In: 2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). pp. 87–96. IEEE (2017)
4. Bajwa, I.S., Lee, M.G., Bordbar, B.: Sbvr business rules generation from natural language specification. In: AAAI: AI for Business Agility. pp. 2–8 (2011)
5. Bazhenova, E., Zerbato, F., Oliboni, B., Weske, M.: From BPMN process models to DMN decision models. Inf. Syst. **83**, 69–88 (2019)
6. Brand, D., Zafiropulo, P.: On communicating finite-state machines. Journal of the ACM (JACM) **30**(2), 323–342 (1983)
7. Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics, analysis and simplification of DMN decision tables. Inf. Syst. **78**, 112–125 (2018)
8. Davis, J.C., Coghlan, C.A., Servant, F., Lee, D.: The impact of regular expression denial of service (redos) in practice: an empirical study at the ecosystem scale. In: Proceedings of the 2018 26th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering. pp. 246–256 (2018)
9. Endres, C., Breitenbücher, U., Falkenthal, M., Kopp, O., Leymann, F., Wettinger, J.: Declarative vs. imperative: two modeling patterns for the automated deployment of applications. In: Proceedings of the 9th International Conference on Pervasive Patterns and Applications. pp. 22–27. Xpert Publishing Services (XPS) (2017)
10. Guanciale, R., Tuosto, E.: Pomcho: A tool chain for choreographic design. Sci. Comput. Program. **202**, 102535 (2021)
11. Hasic, F., Vanthienen, J.: Complexity metrics for DMN decision models. Comput. Stand. Interfaces **65**, 15–37 (2019)
12. Honda, K., Yoshida, N., Carbone, M.: Multiparty asynchronous session types. J. ACM **63**(1), 9:1–9:67 (2016)
13. Kapos, G., Tsadimas, A., Kotronis, C., Dalakas, V., Nikolaidou, M., Anagnostopoulos, D.: A declarative approach for transforming sysml models to executable simulation models. IEEE Trans. Syst. Man Cybern. Syst. **51**(6), 3330–3345 (2021)
14. Kohar, R., Parimala, N.: A metrics framework for a WS-CDL process under evolution. Int. J. Syst. Assur. Eng. Manag. **11**(5), 865–882 (2020)
15. Lee, E., Kim, Y.G., Seo, Y.D., Seol, K., Baik, D.K.: Ringa: Design and verification of finite state machine for self-adaptive software at runtime. Information and Software Technology **93**, 200–222 (2018)
16. Levy, F., NazarenkoF, A.: Formalization of natural language ´ regulations through sbvr structured english - (tutorial). In: Theory, Practice, and Applications of Rules on the Web - 7th International Symposium, RuleML. pp. 19–33 (2013)
17. Manaf, N.A., Antoniades, A., Moschoyiannis, S.: SBVR2Alloy: An SBVR to Alloy compiler. In: 10th IEEE Conference on Service-Oriented Computing and Applications, SOCA 2017. pp. 73–80. IEEE Computer Society (2017)
18. Manaf, N.A., Moschoyiannis, S., Krause, P.J.: Service choreography, sbvr, and time. In: Proc. 14th International Workshop on Foundations of Coordination Languages and Self-Adaptive Systems, FOCLASA. EPTCS, vol. 201, pp. 63–77 (2015)
19. Mansour, K.S., Hammal, Y.: Atl based refinement of ws-cdl choreography into bpel processes. In: International Symposium on Modelling and Implementation of Complex Systems. pp. 329–343. Springer (2018)
20. Marinos, A., Gazzard, P., Krause, P.J.: An SBVR Editor with Highlighting and Auto-Completion. In: Semantic Web Rules - International Symposium, RuleML. pp. 111–118 (2011)

21. Marinos, A., Moschoyiannis, S., Krause, P.: Towards a RESTful infrastructure for Digital Ecosystems. International Journal of Electronic Business **9** (2011)
22. Montali, M.: Specification and verification of declarative open interaction models: a logic-based approach, vol. 56. Springer Science & Business Media (2010)
23. Montali, M., Pesic, M., Aalst, W.M.v.d., Chesani, F., Mello, P., Storari, S.: Declarative specification and verification of service choreographiess. ACM Transactions on the Web (TWEB) **4**(1), 1–62 (2010)
24. Moschoyiannis, S., Krause, P.J.: True concurrency in long-running transactions for digital ecosystems. Fundamenta Informaticae (2015)
25. Moschoyiannis, S., Maglaras, L.A., Manaf, N.A.: Trace-based verification of rule-based service choreographies. In: 11th IEEE Conference on Service-Oriented Computing and Applications, SOCA 2018, Paris, France, November 20-22, 2018. pp. 185–193. IEEE Computer Society (2018). https://doi.org/10.1109/SOCA.2018.00034, https://doi.org/10.1109/SOCA.2018.00034
26. OMG: Business Process Model and Notation (BPMN), vol. Version 2.0. OMG document formal/2011-01-03, http://www.omg.org/spec/BPMN/2.0/
27. OMG: Decision Model and Notation (DMN), vol. Version 1.3. OMG document formal/2021-01-01, https://www.omg.org/spec/DMN
28. OMG: Semantics of Business Vocabulary and Business Rules (SBVR), vol. Version 1.5. OMG document formal/dtc/2019-10-02, https://www.omg.org/spec/SBVR/1.5/PDF
29. OMG: Unified Modeling Language (UML), vol. Version 2.5.1. OMG document formal/2017-12-05, https/www.omg.org/spec/UML/
30. OMG: Date-Time Vocabulary (DTV), Version 1.3. OMG document formal/dtc/2016-02-20, http://www.omg.org/spec/DTV/1.3/Beta2 (2017)
31. Pandit, D., Bansal, A.: A declarative approach for an adaptive framework for learning in online courses. In: Getov, V., Gaudiot, J., Yamai, N., Cimato, S., Chang, J.M., Teranishi, Y., Yang, J., Leong, H.V., Shahriar, H., Takemoto, M., Towey, D., Takakura, H., Elçi, A., Takeuchi, S., Puri, S. (eds.) 43rd IEEE Annual Computer Software and Applications Conference, COMPSAC 2019, Milwaukee, WI, USA, July 15-19, 2019, Volume 1. pp. 212–215. IEEE (2019)
32. Papazoglou, M.P., Georgakopoulos, D.: Introduction: Service-oriented computing. Commun. ACM **46**(10), 24–28 (2003)
33. Sipser, M.: Introduction to the Theory of Computation. Cengage learning (2012)
34. Testa, O.A., C., E.R.F., Montejano, G., Debnath, N.C., Dieste, O.: WS-CDL: coordinating ubiquitous devices in pervasive environments using a web standard. In: 2020 IEEE International Conference on Industrial Technology, ICIT 2020, Buenos Aires, Argentina, February 26-28, 2020. pp. 1007–1012. IEEE (2020). https://doi.org/10.1109/ICIT45562.2020.9067311, https://doi.org/10.1109/ICIT45562.2020.9067311
35. Van Der Aalst, W.M., Pesic, M.: Decserflow: Towards a truly declarative service flow language. In: International Workshop on Web Services and Formal Methods. pp. 1–23. Springer (2006)
36. W3C: Web Services Choreography Description Language (WS-CDL). W3C Working Group, http://www.w3.org/TR/ws-cdl-10-primer/ (2006)
37. Wild, K., Breitenbücher, U., Képes, K., Leymann, F., Weder, B.: Decentralized cross-organizational application deployment automation: An approach for generating deployment choreographies based on declarative deployment models. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) Advanced Information Systems Engineering - 32nd International Conference, CAiSE 2020, Grenoble,

France, June 8-12, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12127, pp. 20–35. Springer (2020)