Automation of Workflow Design in an Industrial Enterprise

Nikolay Voit, Sergey Kirillov(✉), and Dmitry Kanev

Ulyanovsk State Technical University, Ulyanovsk, Russia
n.voit@ulstu.ru, kirillovsyu@gmail.com,
dima.kanev@gmail.com

Abstract. The fundamental scientific problem of the business process management theory is to increase the efficiency of automated systems workflow synthesis and processing in order to reduce the time spent on their development, increasing the success of processing diagram models, namely the implementation of the requirements for resource constraints, functionality, financial component and deadlines, as well as improving the diagram models quality in terms of error control, narrowing the semantic gap between business process analysis and execution. The article proposes an approach to the analysis of workflow diagram models on the basis of temporal automatic grammar with linear analysis time. The approach allows to control and analyze structural-semantic and temporal errors. The results of the research represent that the approach has significant advantages over similar methods of analysis. The effectiveness of this analytical approach is proved by concrete real and relevant examples.

Keywords: Workflow · Business-process · Grammar · Visual language

1 Introduction

The industrial enterprises success in the market depends on many factors: the range of offered services, market saturation, marketing policy, etc. In order to maintain competitiveness, modern, focused on continuous development, industrial enterprises are forced to constantly improve their activities, which requires the development of new technologies and business techniques, as well as the introduction of more effective management methods and activities organization. That is why, among other activities, it is necessary to be able to choose and use the methodology of design, modeling and analysis (processing) of business processes. Today, methodologies and tools for processing business processes are both a serious area of research and a thriving sector of the software market. The range of processing methods used to describe business processes is very wide: from the simplest graphical notations used to construct flowcharts of algorithms, and such strict mathematical devices as Petri nets, to object-oriented modeling languages such as UML (Unified Modeling Language), and specially designed to describe business systems methodologies, such as XPDL (XML Process Definition Language) and BPEL (Business Process Execution Language).

Analysis methods are used to research the qualitative and quantitative characteristics of workflows. Qualitative characteristics are understood as logical and algebraic

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11623, pp. 551–561, 2019.
https://doi.org/10.1007/978-3-030-24308-1_44

correctness of workflows, formalized with the graph theory, workflow networks, matching matrices, graphical modeling languages, including UML, BPMN, IDEF0 and eEPC, etc., as well as evolutionary approach, logic of statements, etc. Quantitative characteristics represent the effectiveness of the workflows execution in the parameters, such as average service time, the utilization rate of production capacity (downtime), etc. Evaluation of the workflows effectiveness is done using simulation modelling (Petri nets), Markov chains and Queuing theory (Queuing systems). The use of a time machine in the workflows design, specification, control and analysis in the development of complex technical systems in an industrial enterprise is a well-known practice [1]. Time and hybrid automata are used for workflows analysis and management [2] in solving problems of access to resources, blocking, limitation of liveliness (vitality, reversibility, boundedness, reachability, dead transitions, deadlocks, home states). For example, it can be tasks such nuclear reactor temperature control, barrier control at the railway crossing [3], in which time context-free grammars are successfully applied. The large number presence of interacting complex automated systems poses the problem of formal control and analysis that can be performed by various methods. Currently π-calculus is a promising, but still very young and developing theory, it has many open questions and unsolved problems. Widely used Petri nets do not have a universal framework for workflows modeling and analysis. In order to analyze various properties (liveliness, reachability, safety) of workflows are modeled in different types of Petri nets. The model checking method is widely used for the workflows analysis in the development of error-free systems at the conceptual design stage. However, it is intended for experienced scientists and engineers, as it is difficult to understand and use [2]. Workflows are also specified by managers who are not trained in formal models and Informatics, and formal analysis requires a detailed representation of the process model in a formal language that is difficult for managers to construct and understand.

Thus, the problem of the workflows analysis and management mechanisms researching is relevant and has great practical importance. The authors propose a mathematical apparatus based on the development of temporal automatic RVTI-grammar. The work is focused on the automation of workflows qualitative structural, semantic and temporal errors identification. The work contains an Introduction, Related Work, the theoretical statements and Conclusion.

2 Related Work

The theory of business process management engaged in both business practice and science. There are areas of business processes design, analysis, modeling, implementation and control [4–19]. In the modern theory of business processes represented with graphic visual languages, the logical model (behavioral model) is used [20, 21], which contain graphic objects and connections between them. The following graphic languages are widely used at large enterprises: UML [22], BPMN [23], AMBER [19], IDEF [24], eEPC [25], PERT [26]. The paper [19] describes the AMBER language, which has a simple data structure (there are no arrays, records and classes), so the implementation of complex business process structures is impossible. The structural approach, laid down in the methodology IDEF [24], was developed in the languages

UML, BPMN, eEPC and specialized language Pilot Workflow ASCON (Russian developer of workflow management systems, firm ASCON [27]) in terms of the object-oriented paradigm inheritance and the introduction of the "time" concept in the diagram workflow models [2]. However, in the most common tools for creating and processing diagram models, such as Microsoft Visio [28], Visual paradigm for UML [29], Aris Toolset [25], IBM Rational Software Architect (RSA) [30], the Pilot workflow ASCON analysis of diagram models is performed by direct methods, requires several "passes" depending on the type of error being controlled, no analysis of integrated diagrammatically models structural peculiarities and operating semantic analysis attached software modules diagrammatically models of dynamic workflows.

Promising approach for treatment diagrammatically workflow is syntactically oriented and based on formal grammars. The most well-known are Web grammar [31], Positional grammar [32], Relational grammar [33], multilevel graph grammar [34] and preserving graph grammar [35]. Positional grammars are the simplest. Developing on the basis of Plex structures [34], they inherited their shortcomings. These grammars do not involve the use of join regions. They cannot be used for graphic languages whose graphical objects have a dynamically changing number of inputs/outputs and cannot be used to control the syntax of graphical languages that contain parallelism. The advantage of relational grammars is the ability to handle errors, but they do not have a mechanism to neutralize such errors. Multilevel and preserving graph grammars are able to provide the graphic languages analysis with "deep" context dependence, which is necessary in the languages, allowing to specify synchronization of the performed actions. Examples of such languages are the languages of process flow diagrams and graphical message sequence Charts. Common shortcomings [36–39] of these grammars are:

1. Increasing the number of products in the construction of grammar for unstructured graphic languages, i.e., with a constant number of graphic language primitives, there is a significant increase in the number of products, because it is necessary to determine all possible variants of unstructured.
2. The grammar construction complexity (increasing the complexity of products and their number), and for some formalisms the impossibility of grammar construction, for graph-schemes with unstructured parallelism.
3. High time complexity. The analyzers built on the basis of the considered grammars offer polynomial or exponential analysis time of graphic languages diagrams.

The main limitation of the above-mentioned methods is that they do not work in the presence of different types (temporal, multi-level, etc.) of diagrams simultaneously, which means that in some cases the input diagrams cannot be analyzed.

3 Workflow Analysis on the Example of the Design and Approval Process of Engineering Documentation

The main approach is to use special RV-grammar family for analyzing, translating, synthesis and control of business processes.

Table 1 presents a mathematical description of the author's RVTI-grammar for graphic language BPMN, proposed in [40–42].

Table 1. Grammar for basic BPMN

Prev. state	Quasi termin	Next state	Memory operations
r0	A0	r1	$W_1(t^{1m^{(n-1)}})$
r1	rel	r2	Θ
r2	E	r1	$W_1(1^{t(1)}, 1^{t(4)}, (ts_{max} + tc_{)}^{t(15)})/W_3(e^{t(1)})$
	Em	r1	$W_1(1^{t(2)}, 1^{t(5)}, (ts_{max} + tc_{)}^{t(15)}))/W_3(e^{t(5)})$
	Et	r1	$W_1(1^{t(3)}, 1^{t(6)}, (ts_{max} + tc_{)}^{t(15)})/W_3(e^{t(6)})$
	Ak	r3	Θ
	A	r4	$W_1(1^{t(16)}, k^{t(17)}, t^{1m^n}, (ts_{max} + tc_{)}^{t(15)})/W_3(e^{t(17)})$
	_A	r4	$W_1(inc(m^{t(16)}), comp(ts_{max} + tc)^{t(15)})/W_3(m^{t(16)} < n^{t(17)})$
	EG	r5	$W_1(1^{t(9)},k^{t(10)}, t^{1m^n}, (ts_{max} + tc_{)}^{t(15)})/W_3(e^{t(10)})$
	_EG	r5	$W_1(inc(m^{t(9)}), comp(ts_{max} + tc)^{t(15)})/W_3(m^{t(9)} < n^{t(10)})$
	EBG	r6	$W_1(1^{t(11)}, k^{t(12)}, t^{1m^n}, (ts_{max} + tc_{)}^{t(15)})/W_3(e^{t(12)})$
	_EBG	r6	$W_1(inc(m^{t(11)}), comp(ts_{max} + tc)^{t(15)})/W_3(m^{t(11)} < n^{t(12)})$
	PG	r7	$W_1(1^{t(13)},k^{t(14)}, t^{1m^n}, (ts_{max} + tc_{)}^{t(15)})/W_3(e^{t(14)})$
	_PG	r7	$W_1(inc(m^{t(13)}), comp(ts_{max} + tc)^{t(15)})/W_3(m^{t(13)} < n^{t(14)})$
r3	labelA0	r2	$W_2(b^{1m}, tc^{t(15)})$
	labelA	r2	$W_2(b^{2m}, tc^{t(15)})$
	labelEG	r2	$W_2(b^{3m}, tc^{t(15)})$
	labelEBG	r8	$W_2(b^{4m}, tc^{t(15)})$
	labelPG	r2	$W_2(b^{5m}, tc^{t(15)})/W_3(m^{t(13)} = n^{t(14)})$
	no_label	r9	*
r4	labelA0	r2	$W_2(b^{1m}, tc^{t(15)})$
	labelA	r2	$W_2(b^{2m}, tc^{t(15)})$
	labelEG	r2	$W_2(b^{3m}, tc^{t(15)})$
	labelEBG	r8	$W_2(b^{4m}, tc^{t(15)})$
	Et	r1	$W_1(1^{t(3)}, 1^{t(6)}, (ts_{max} + tc_{)}^{t(15)})/W_3(e^{t(6)})$
r5	labelA0	r2	$W_2(b^{1m}, tc^{t(15)})$
	labelEG	r2	$W_2(b^{3m}, tc^{t(15)})$
	labelEBG	r8	$W_2(b^{4m}, tc^{t(15)})$
r6	labelA0	r2	$W_2(b^{1m}, tc^{t(15)})$
	labelEG	r2	$W_2(b^{3m}, tc^{t(15)})$
	labelEBG	r8	$W_2(b^{4m}, tc^{t(15)})$
	relEBG	r8	Θ
r7	labelA	r2	$W_2(b^{2m}, tc^{t(15)})$
	labelPG	r2	$W_2(b^{5m}, tc^{t(15)})/W_3(m^{t(13)} = n^{t(14)})$
r8	Em	r1	$W_1(1^{t(2)}, 1^{t(5)}, (ts_{max} + tc_{)}^{t(15)}))/W_3(e^{t(5)})$
r9			

In Table 1, “n” is the number of outgoing links; “k” is the number of incoming links; “m” is the number of analyzed incoming links; “t” is the number of the current graphic primitive from the number of primitives of this type; “b” is the number of the

graphic primitive from which the "control signal" comes (applies only to links); "e" is the sign of emptiness; "tsmax" is the maximum time characteristic of the operation, "tc" is the current time characteristic; "inc()" is the operation of the value increment for the current element; "comp()" is the operation returns a larger value between the current value and what is already written in the tape for the current element.

For example, the operation "W1(1t(1), kt(2), $t^{1m^{(n-1)}}$)/W3(et(2))" can be described as follows: provided that the tape number "2" value for the current element is not set, the number "1" is recorded in the tape number "1" for the current element, the total number of incoming links k in the tape number "2" for the current element and the number of element t in the store number "1" in quantity (n − 1).

Symbol "∗" implies the following operations: * = W2(e1 m) … && W3(mai (7) == nai(8)) … && W3(mai(8) > 1) … && W3(mai(15) < td)…, where «td» is the time characteristic of the deadline.

An operation of the form "W2(e1 m)" means that all stores must be empty (i.e. there are no return points)."W2(1ai(1))" is operation of reading units from all cells of tape #1 to check elements with 1 fixed link; "W3(mai(1) = nai(2))" is operation of checking whether the number of analyzed links corresponds to the total number of corresponding element inputs; "W3(mai(8) > 1)" is operation of checking the number of incoming links that they are greater than one; "W3(mai(15) < td)" is operation of checking the output of time characteristics for a certain period of time;

The automaton of temporal automata-based RVTI grammar [43, 44] of visual BPM language is presented as a graph in Fig. 1.

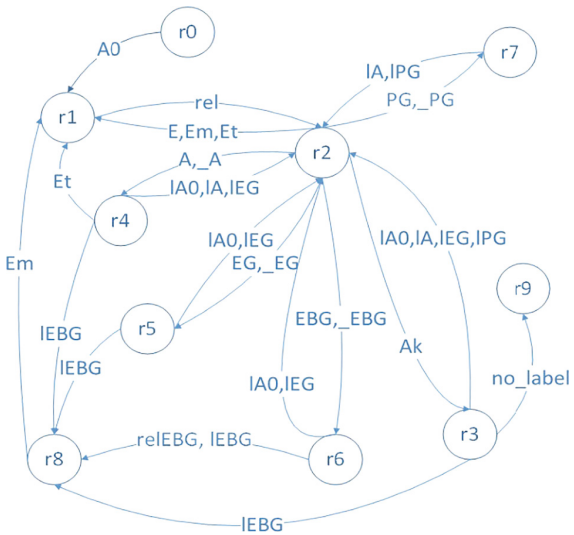

Fig. 1. Temporal automata-based RVTI grammar of the visual language BPMN presented as a graph

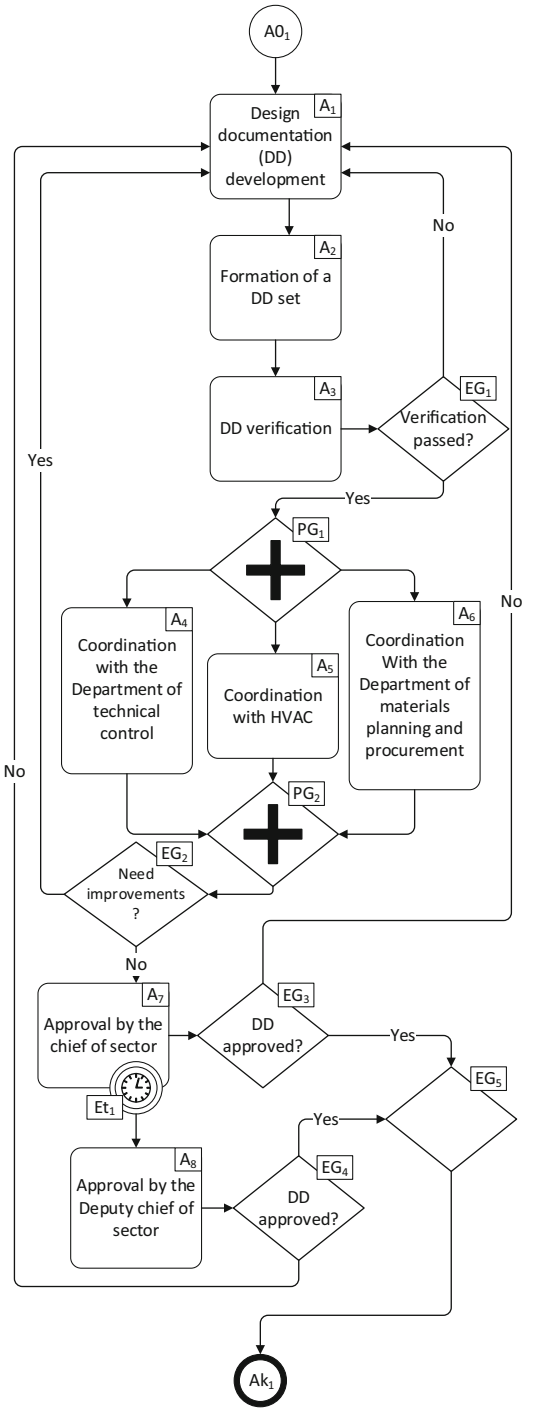

Fig. 2. Design documentation development process, presented on BPMN

We decided to test the developed mathematical apparatus and methods on the real business process of a large industrial enterprise. The process of design documentation development is shown in Fig. 2. It should be borne in mind that in the business process there are cycles, so in the most "sad" situation, he can strive for endless implementation. The exact value can only be obtained when the real process is started. In a situation where the process has a certain completion date and the analysis is static, the maximum time limit is assumed. If the process does not have time to complete in the absence of design documentation (DD) revision, it will not be able to complete if they are available, so the calculation is carried out for the first option. Therefore, it is necessary to pre-build a business process in advance in a more optimal way, to reduce the amount of execution of certain tasks or to postpone the deadlines forcibly.

To demonstrate the grammar's ability to control semantic errors, the correct example was changed to erroneous by reassigning the endpoint to PG1 connections from EG3 to A1. This change will result in a logical error called a "deadlock". At the moment, the memory stores only one continuator, and the link is coming from element 1 PG. However, the retrieval of a continuation is impossible because it fails the necessary condition (1 = 2) analysis of all incoming links (Table 2).

Table 2. State of machine memory at the time of the "deadlock" error

Step	Quasi termin	Memory operations	State of machine memory			Errors
11.	labelPG	$W_2(b^{1m})/W_3(m^{t(1)} == n^{t(2)})$	1 m	t(1)	t(2)	Deadlock
			PG_1	1_{PG1}	2_{PG1}	

The developed software that implements the described methods successfully catches structural, semantic and temporal types of errors [45, 46]. The screenshot of the error message of the computer program for the analysis of the diagram model of the visual language BPM is shown in Fig. 3.

An experiment was conducted to confirm the linear velocity of the method. To create a test base of diagram models, students were involved, as well as a special software tool that allows you to generate voluminous business processes. For the purity of the experiment, analysis and control were performed in several iterations for each diagram. The generation mechanisms allowed to take into account the main criteria that most significantly affect the final measurement time, namely the number of elements and errors, their complexity and the maxima of the "fork" (Fig. 4).

With various combinations of these criteria values, a General trend is observed on the chart, which confirms the theoretical calculations about the linear nature of the speed of the method [43].

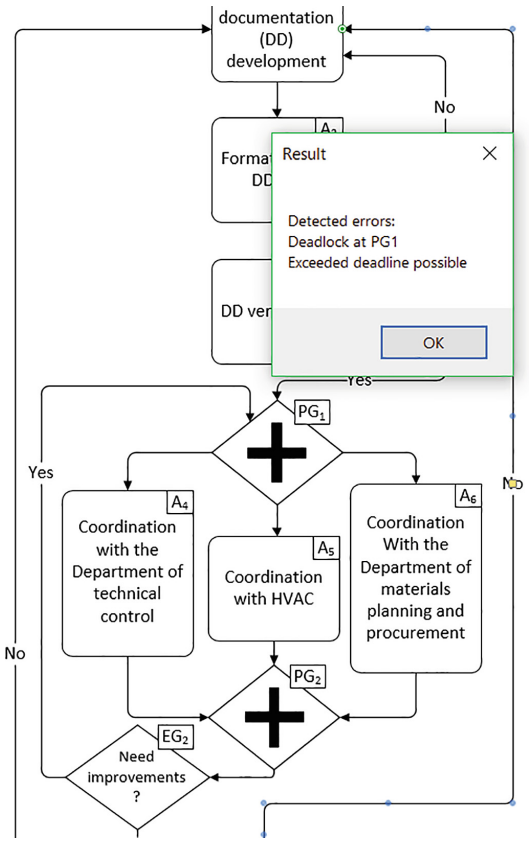

Fig. 3. Error screenshot, found by the developed computer program

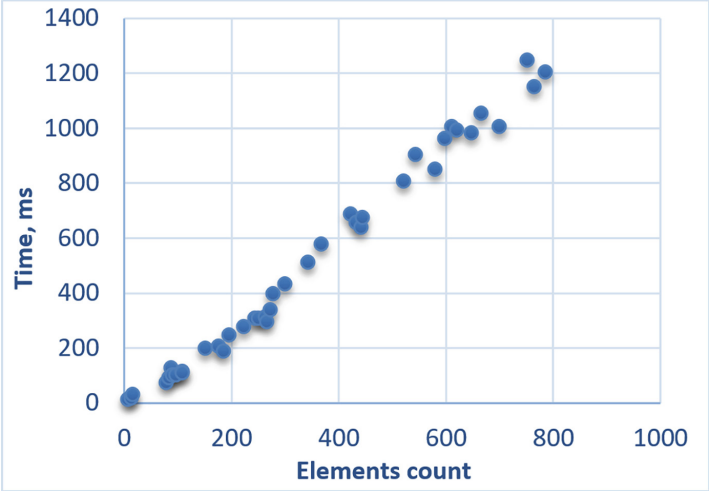

Fig. 4. A dependence graph of the analysis time by the automatic RVTI grammar on the number of elements

4 Conclusion and Future Works

Developed automata-based temporal grammar for visual language BPMN. A special feature is the linear characteristic of the time of the algorithm. The initial experiment showed practical confirmation of this. In future works, it is expected to attract more test data, both through automatic generation and manual generation business processes diagrams, to conduct a more detailed experiment, including comparison with the analogues. Work is also underway to expand the possibility of semantic analysis with respect to the textual matching of the diagrams attributes of the same processes in different visual languages between themselves and the project documentation.

Acknowledgements. The reported study was funded by RFBR according to the research project № 17-07-01417 and Russian Foundation for Basic Research and the government of the region of the Russian Federation, grant № 18-47-730032.

References

1. Lee, E.A.: Cyber physical systems: design challenges. In: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC) (2008)
2. Wang,Y., Fan, Y.: Using temporal logics for modeling and analysis of workflows. In: IEEE International Conference on E-Commerce Technology for Dynamic E-Business (2004)
3. Heitmeyer, C., Lynch, N.: The generalized railroad crossing: a case study in formal verification of real-time systems. In: Proceedings Real-Time Systems Symposium REAL-94 (1994)
4. Schael, T.: Workflow Management Systems for Process Organisations. LNCS, vol. 1096. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-49450-2
5. Van Der Aalst, W.M.P., et al.: Ex*Spect* 6.4 an executable specification tool for hierarchical colored petri nets. In: Nielsen, M., Simpson, D. (eds.) ICATPN 2000. LNCS, vol. 1825, pp. 455–464. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44988-4_26
6. Dellarocas, C., Klein, M.: A knowledge-based approach for designing robust business processes. In: van der Aalst, W., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 50–65. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45594-9_4
7. Sharp, A., McDermott, P.: Workflow Modeling: Tools for Process Improvement and Applications Development. Artech House, Norwood (2009)
8. Bock, C.: Introduction to business process and definition metamodel. US National Institute of Standard and Technology. Manufacturing Engineering (2008)
9. Poizat, P., Salaün, G., Krishna, A.: Checking business process evolution. In: Kouchnarenko, O., Khosravi, R. (eds.) FACS 2016. LNCS, vol. 10231, pp. 36–53. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57666-4_4
10. Martens, A.: Analyzing web service based business processes. In: Cerioli, M. (ed.) FASE 2005. LNCS, vol. 3442, pp. 19–33. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31984-9_3
11. Raedts, I., Petkovic, M., Usenko, Y.S., van der Werf, J.M.E., Groote, J.F., Somers, L.J.: Transformation of BPMN models for behaviour analysis. In: Proceedings of the 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (2007)

12. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in BPMN. Inf. Softw. Technol. **50**(12), 1281–1294 (2008)
13. Wong, P.Y.H., Gibbons, J.: A process semantics for BPMN. In: Liu, S., Maibaum, T., Araki, K. (eds.) ICFEM 2008. LNCS, vol. 5256, pp. 355–374. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88194-0_22
14. Decker, G., Weske, M.: Interaction-centric modeling of process choreographies. Inf. Syst. **36** (2), 292–312 (2011)
15. Decker, G., Weske, M.: Local enforceability in interaction petri nets. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 305–319. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_22
16. Güdemann, M., Poizat, P., Salaün, G., Dumont, A.: VerChor: a framework for verifying choreographies. In: Cortellessa, V., Varró, D. (eds.) FASE 2013. LNCS, vol. 7793, pp. 226–230. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37057-1_16
17. Kossak, F., et al.: A rigorous semantics for BPMN 2.0 process diagrams. In: Kossak, F., et al. (eds.) A rigorous semantics for BPMN 2.0 process diagrams, pp. 29–152. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09931-6_4
18. Janssen, W., Mateescu, R., Mauw, S., Springintveld, J.: Verifying business processes using SPIN. In: Proceedings of the 4th International SPIN Workshop, November, pp. 21–36 (1998)
19. Van Der Aalst, W., Van Hee, K.M., van Hee, K.: Workflow Management: Models, Methods, and Systems. MIT press, Cambridge (2004)
20. Fischer, L. (ed.): Workflow Handbook 2005. Workflow Management Coalition (2005)
21. Booch, G., Jacobson, I., Rumbaugh, J.: The unified modeling language reference manual (1999)
22. Business Process Model and Notation (BPMN), v. 2.0. OMG (2011). http://www.omg.org/spec/BPMN/2.0
23. Mayer, R.J., Painter, M.K., de Witte, P.S.: IDEF Family of Methods for Concurrent Engineering and Business Re-Engineering Applications. Knowledge Based Systems, College Station (1994)
24. Santos, P.S., Almeida, J.P.A., Pianissolla, T.L.: Uncovering the organisational modelling and business process modelling languages in the ARIS method. Int. J. Bus. Process Integr. Manag. **5**(2), 130 (2011)
25. Pozewaunig, H., Eder, J., Liebhart, W.: ePERT: extending PERT for workflow management systems. In: ADBIS, September, pp. 217–224 (1997)
26. https://ascon.ru
27. Roth, C.: Using Microsoft Visio 2010. Pearson Education, London (2011)
28. Visual Paradigm: Visual Paradigm for UML. Visual Paradigm International, Hong Kong (2010)
29. Hoffmann, H.-P.: Deploying model-based systems engineering with IBM® rational® solutions for systems and software engineering. In: 2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC) (2012)
30. Suppes, P.: Syntactic methods in pattern recognition (K. S. Fu). SIAM Rev. **19**(4), 746 (1977)
31. Costagliola, G., De Lucia, A., Orefice, S., Tortora, G.: Positional grammars: a formalism for LR-like parsing of visual languages. In: Marriott, K., Meyer, B. (eds.) Visual Language Theory, pp. 171–191. Springer, New York (1998). https://doi.org/10.1007/978-1-4612-1676-6_5
32. Zhang, D.-Q., Zhang, K.: Reserved graph grammar: a specification tool for diagrammatic VPLs. In: Proceedings of the 1997 IEEE Symposium on Visual Languages (1997). (Cat. No. 97TB100180)

33. Rekers, J., Schürr, A.: Defining and parsing visual languages with layered graph grammars. J. Vis. Lang. Comput. **8**(1), 27–55 (1997)
34. Zhang, D.-Q.: A context-sensitive graph grammar formalism for the specification of visual languages. Comput. J. **44**(3), 186–200 (2001)
35. Sharov, O.G., Afanas'ev, A.N.: Syntax-directed implementation of visual languages based on automaton graphical grammars. Program. Comput. Softw. **31**(6), 332–339 (2005)
36. Sharov, O.G., Afanas'ev, A.N.: Methods and tools for translation of graphical diagrams. Program. Comput. Softw. **37**(3), 171–179 (2011)
37. Aho, A.V.: Compilers: Principles, Techniques and Tools (for Anna University), 2nd edn. Pearson Education, Noida (2003)
38. Sharov, O.G., Afanasiev, A.N.: Syntax error recovery in graphical languages. Program. Comput. Softw. **34**(1), 44–48 (2008)
39. Afanasyev, A., Voit, N.: Grammar-algebraic approach to analyze workflows. In: Gervasi, O., et al. (eds.) ICCSA 2018. LNCS, vol. 10963, pp. 499–510. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95171-3_39
40. Afanasyev, A., Voit, N., Timofeeva, O., Epifanov, V.: Analysis and control of hybrid diagrammatical workflows. In: Abraham, A., Kovalev, S., Tarassov, V., Snasel, V., Vasileva, M., Sukhanov, A. (eds.) IITI 2017. AISC, vol. 679, pp. 124–133. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-68321-8_13
41. Afanasyev, A., Ukhanova, M., Ionova, I., Voit, N.: Processing of design and manufacturing workflows in a large enterprise. In: Gervasi, O., et al. (eds.) ICCSA 2018. LNCS, vol. 10963, pp. 565–576. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95171-3_44
42. Voit, N.N.: Development of timed RT-grammars for analysis of business process at manufacturing and in cyber-physical systems. In: 2017 International Conference on Computing Networking and Informatics (ICCNI) (2017)
43. Afanasyev, A.N., Voit, N.N., Kirillov, S.Y.: Development of RYT-grammar for analysis and control dynamic workflows. In: 2017 International Conference on Computing Networking and Informatics (ICCNI) (2017)
44. Afanasyev, A., Voit, N., Gaynullin, R.: The analysis of diagrammatic models of workflows in design of the complex automated systems. In: Abraham, A., Kovalev, S., Tarassov, V., Snášel, V. (eds.) Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'16). AISC, vol. 450, pp. 227–236. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33609-1_20
45. Afanasyev, A., Voit, N., Gaynullin, R.: The analysis of diagrammatic of workflows in design of the automated systems. In: Uncertainty Modelling in Knowledge Engineering and Decision Making (2016)
46. Afanasyev, A.N., Voit, N.N., Voevodin, E.Y., Gainullin, R.F.: Control of UML diagrams in designing automated systems software. In: 2015 9th International Conference on Application of Information and Communication Technologies (AICT) (2015)