Temporal Automata RVTI-Grammar for Processing Diagrams in Visual Languages as BPMN, eEPC and Askon-Volga

Alexander Nikolaevich Afanasyev
Department of Computer engineering
Ulyanovsk State Technical University
Ulyanovsk, Russia
a.afanasev@ulstu.ru

Nikolay Nikolaevich Voit
Department of Computer engineering
Ulyanovsk State Technical University
Ulyanovsk, Russia
n.voit@ulstu.ru

Sergey Yurievich Kirillov
Department of Computer engineering
Ulyanovsk State Technical University
Ulyanovsk, Russia
kirillovsyu@gmail.com

ABSTRACT
The article proposes an approach to the analysis of grammatical models of work flows on the basis of temporal automatic grammar with linear analysis time. The approach allows to control and analyze structural-semantic and temporal errors in graphic models. The results of the study show 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. Visual languages BPMN, eEPC and ASCON-Volga are used for demonstration.

CCS Concepts
•Theory of computation→ Grammars and context-free languages• Applied computing→ Business process management

Keywords
Business processes, graphical language analysis, BPMN, EPC, workflows

1. INTRODUCTION
Visual languages and special software are actively used in the design of automated systems (AC), as well as other computer facilities, as they help to significantly increase the understanding of the system by participants in the process of creating AC, as well as to increase the speed of development and quality of the created systems by unifying the language of interaction. Additionally, documentation of design and architectural, functional solutions and formal control of correctness of diagrams is simplified. The basis of technical solutions at the moment laid a lot of different theoretical methods, concepts, grammars, etc.

For example, the formal languages section of theoretical computer science represents the flow of work using mathematical tools such as graphic grammars, finite automata (mile, Moore, Byuhi), transition systems (Kripke structure), letters (alphabet), words, and sentences. Graphic formal grammars describe complex structures of syntax and morphology of visual languages, and finite automata allow to model computational processes and are models of simple calculations. They have found application in various industrial fields, for example, in software engineering in the analysis of diagram models of computer programs algorithms, in bioinformatics in the study of the secondary structure of RNA, in cyberphysical systems in the management of the nuclear reactor, in transport in the management of the barrier, etc. The most important aspects in the formal grammar is the control, analysis, metacompilation, conversion, synthesis, simulation and comparative analysis of various families and their properties to improve the quality of the treatment workflow, reducing the time of processing and implementation in computer programs. Testing of computer programs, behavior of complex technical systems (objects) is a difficult (sometimes unsolvable) verification task in theoretical computer science with exponential or polynomial complexity. However, finite state machines, being models of simple calculations, effectively (with a linear dependence of time on the number of analyzed objects) solve such problems, which provides a provably properly working complex systems.

Workflows can be conducted simultaneously in different units and different performers, there were problems of synchronization, resource locking, deadlocks, bottlenecks, etc. The authors propose automatic temporal RVTI-grammar to analyse the temporal features of the workflow.

BPMN, eEPC and KTPR were used as the main notations for the work. The developed RVTI grammar for these graphic specifications is designed to help solve possible problems.

The work contains Introduction, Related Work, RV-grammar Family, Experiment and Conclusion and Future Work. Introduction to the subject area, the study of the subject and the object of research is presented in Introduction. An overview and analysis of the work is presented in the Related Work section. The description of RVTI grammar is presented in the second section of RVTI grammar for processing visual languages. The paper also includes an experiment and evaluation of the effectiveness of the proposed grammars. The General conclusion on the work and further directions of research are contained in the Conclusion and Future Work.

ICCTA 2019, April 16–17, 2019, Istanbul, Turkey

ACM ISBN 978-1-4503-7181-0/19/04…$15.00
https://doi.org/10.1145/3323933.3324067

2. GRAMMARS REVIEW

For processing of graphic visual languages use a graphical grammar. The most well-known are web grammar [1], positional graphic grammar [2,3], relating to context-free grammar, relational graphic grammar [4], preserving graphic grammar [5]. However, the mentioned graphical grammars have the following drawbacks.

1. Positional grammars, developing on the basis of Plex structures, do not involve the use of connection areas and can not be used for graphic languages whose objects have a dynamically changing number of inputs/outputs.

2. The authors of relational grammars stipulate the imperfection of the neutralization mechanism in terms of the incompleteness of the generated error list.

3. There is no control of semantic integrity (text attributes of complex diagram models presented in different visual languages), as well as semantic consistency in terms of structural issues of diagrams between themselves and the conceptual model as a whole.

4. The common drawbacks of the above-described grammars are: an increase in the number of products in the construction of grammar for unstructured graphic languages (with a constant number of the graphic language primitives to describe all variants of unstructured there is a significant increase in the number of products), the complexity of grammar construction, the large time costs of analysis (analyzers built on the basis of the considered grammars have a polynomial or exponential time of graphic diagram analysis.

In the most common tools for creating and processing diagram models, such as Microsoft Visio [6], Visual paradigm for UML [7], Aris Toolset [8], IBM Rational Software Architect (RSA) [9], the analysis and control of diagram models is carried out by direct methods, requires several "passes" depending on the type of error, there is no control of the structural features of complex diagram models, as well as semantic control of the integrity and consistency of the structural and textual attributes of the associated diagram models of dynamic work flows.

Thus, to solve the problems of control and analysis it is necessary to develop a complex system of new models, grammars and means of analysis and control of diagram models of dynamic distributed workflows.

3. RVTI-GRAMMAR FOR VISUAL LANGUAGE PROCESSING

Flows of design works are a powerful tool for analyzing the business processes of the enterprise and contain design tasks that are addressed to specific departments and executors of the enterprise.

Such design works can be performed simultaneously in different structural divisions and different executors, so the issues of synchronization, resource blocking, deadlocks, bottlenecks, etc., encountered in the theory of informatics, arise in the field of business process management enterprise. Such streams of design work can be represented in the temporal basis of "Before", "During", "After", putting the time schedule for the execution of the enterprise's work in accordance with the parameter [10].

Temporal RVT - grammar of a language L (G) is an ordered eight non-empty sets

$$G = (V, \Sigma, \tilde{\Sigma}, C, E, R, T, r_0) \quad (1)$$

where $V = \{v_e, e = \overline{1,L}\}$ is auxiliary alphabet; $\Sigma = \{a_t, t = \overline{1,T}\}$ is terminal alphabet graphic language; $\tilde{\Sigma} = \{\widetilde{at}, t = \overline{1,T}\}$ is quasi terminal alphabet; C is set of clock identifiers; E is the set of temporal relations "Before", "During", "After" (initialization of the clock {c := 0}, relations of the form {c~x}, where x the variable (the identifier of the clock), a is a constant, ~ ϵ{=,<, ≤,>,≥}); $R = \{r_i, i = \overline{0,I}\}$ is grammar G schema (set of names of complexes of products, each complex r_i consists of a subset P_{ij} of products $r_i = \{P_{ij}, j = \overline{1,J}\}$); T ϵ $\{t_1, t_2, \ldots, t_n\}$ is a set of time stamps; $r_0 \epsilon R$ is RV-axiom grammar.

Table 1 presents a mathematical description of the author's RVTI-grammar for graphic language BPMN, proposed in [11, 12, 13].

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; “td” is the time characteristic of the deadline; “ts_{max}” 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 “$W_1(1^{t(1)}, k^{t(2)}, t^{1m^{(n-1)}})/W_3(e^{t(2)})$” 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: * = $W_2(e^{1m})$ … && $W_3(m^{ai(7)} == n^{ai(8)})$ … && $W_3(m^{ai(8)} > 1)$ … && $W_3(m^{ai(15)} < td)$… .

An operation of the form “$W_2(e^{1m})$“ means that all stores must be empty (i.e. there are no return points).“$W_2(1^{ai(1)})$“ is operation of reading units from all cells of tape #1 to check elements with 1 fixed link;“$W_3(m^{ai(1)} = n^{ai(2)})$“ is operation of checking whether the number of analyzed links corresponds to the total number of corresponding element inputs; “$W_3(m^{ai(8)} > 1)$“ is operation of checking the number of incoming links that they are greater than one; “$W_3(m^{ai(15)} < td)$“ is operation of checking the output of time characteristics for a certain period of time;

The automaton of temporal automata-based RVTI grammar of visual BPM language is presented as a graph in Figure 1.

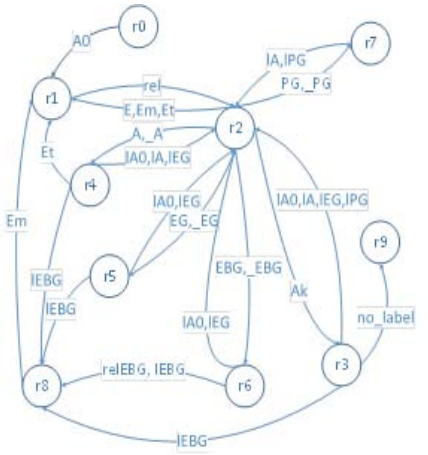

Figure 1. Temporal automata-based RVTI grammar of the visual language BPMN presented as a graph

To demonstrate the grammar's ability to control semantic errors, the correct example has been changed to an erroneous one by reassigning the endpoint to PG_1 of the connection from EG_3 to A_1. This modification will lead to a logical error called “Deadlock“ (see Figure 2 and Figure 3). At the moment, only one continuator is stored in memory and this is the link coming out of the PG_1 element (see Table 2). However, the extraction of this continuator is not possible because the necessary condition ($1 \neq 2$) that all incoming links are analyzed is not met.

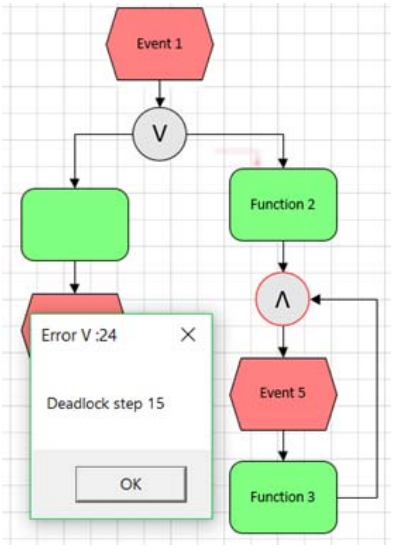

Figure 2. Deadlock error on the eEPC diagram catched by automata-based RVTI grammar program

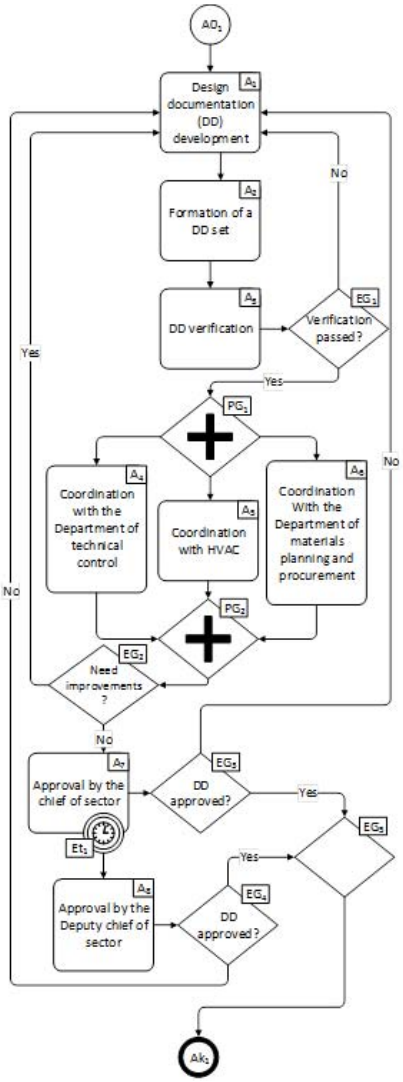

Figure 3. Error screenshot, found by the developed computer program

Table 2. Grammar for basic BPMN.

Step	Quasi termin	Memory operations	State of machine memory			Errors
11.	labelPG	$W_2(b^{1m})/W_3(m^{t(1)} == n^{t(2)})$	1m	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. 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 Figure 4.

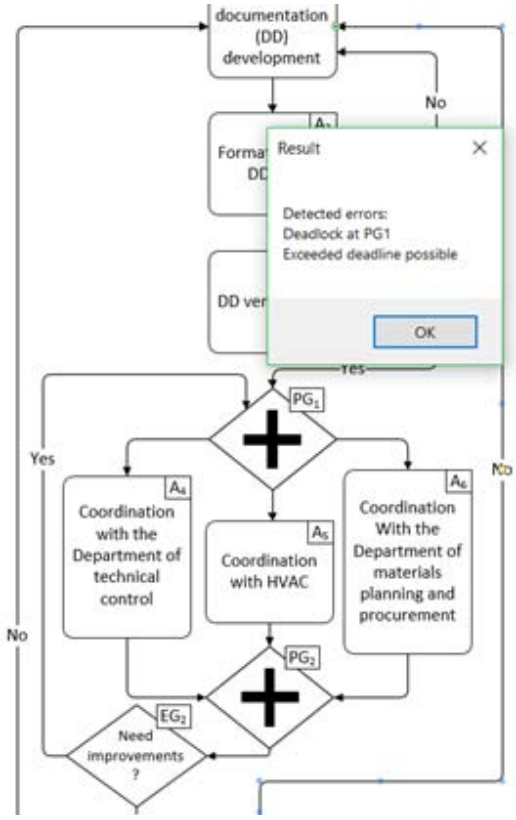

Figure 4. Error screenshot, found by the developed computer program

An experiment was conducted to confirm the linear velocity of the method (see Figure 5). To create a test base of diagram models, students were involved, as well as a special software tool that allows to generate voluminous business processes. For the purity of the experiment, the analysis and control was performed in several iterations for each diagram. The generation mechanisms allowed to take into account the main factors that most significantly affect the final time of measurements, which were the number of elements and errors, their complexity and the maximum "fork".

With various combinations of these factors, a General trend is observed on the chart, which confirms the theoretical calculations about the linear nature of the speed of the method [9, 14].

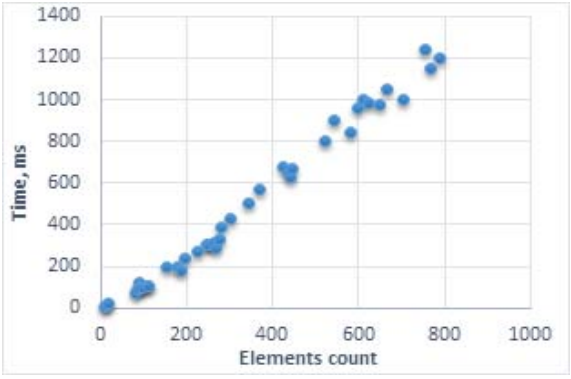

Figure 5. A dependence graph of the analysis time by the automatic RVTI grammar on the number of elements

4. Conclusion

The temporal RVTI grammar proposed by the authors has a linear characteristic of the time of workflows analysis, consider the language of the process description and can be applied to any diagram. It allows you to take into account the time parameters. This will help to detect logical planning errors in the development and design of automated systems.

Further directions of work are the expansion of the possibilities of semantic analysis of diagrammatic models from the point of view of coordinating text attributes of diagrams with project documentation.

5. ACKNOWLEDGMENTS

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.

6. REFERENCES

[1] Fu, K. 1977. *Structural methods of pattern recognition*. Mir, Moscow, SU.

[2] Zhang, D.-Q. and Zhang, K. 1997. Reserved graph grammar: A specification tool for diagrammatic VPLs. Visual Languages. In *Proceedings of 1997 IEEE Symposium on Visual Languages* (Isle of Capri, Italy, Sept. 23-26, 1997). IEEE Computer Society, Washington, DC, USA, 284–291. DOI= https://doi.org/10.1109/VL.1997.626596

[3] Costagliola, G., Lucia, A.D., Orece, S., and Tortora, G. 1997. A parsing methodology for the implementation of visual systems. In *IEEE Transactions on Software Engineering*, 23, 12 (Dec. 1997), 777–799. DOI= http://doi.org/10.1109/32.637392

[4] Wittenburg, K. and Weitzman, L., 1998. Relational grammars: Theory and practice in a visual language interface for process modeling. In: *Visual Language Theory*, K. Marriott, B. Meyer, Eds. Springer, New York, NY, 193-217. DOI= http://doi.org/10.1007/978-1-4612-1676-6_6

[5] Zhang, K. B., Zhang, K., and Orgun, M. A. 2002. Using Graph Grammar to Implement Global Layout for A Visual Programming Language Generation System. In: *Conferences in Research and Practice in Information Technology*, Feng D., Jin J., Eades P., Yan H, Eds. Australian Computer Society, Sydney, 115-122. DOI=

[6] Roth, C. 2011. *Using Microsoft Visio 2010*. Pearson Education, London, UK.

[7] Visual paradigm for uml //Hong Kong: Visual Paradigm International. https://www.visual-paradigm.com/

[8] Santos Jr., P. S., Almeida, J. P. A., and Pianissolla, T. L. 2011. Uncovering the organisational modelling and business process modelling languages in the ARIS method. In *International Journal of Business Process Integration and Management*, 5, 2 (May. 2011), 130–143. DOI= http://doi.org/10.1504/IJBPIM.2011.040205

[9] Afanasyev, A. N., Voit, N. N., Voevodin, E. Y., and Gainullin, R. F. 2015. Control of UML diagrams in designing automated systems software. In *9th International Conference on Application of Information and Communication Technologies (AICT)* (Rostov on Don, Russia). IEEE, 285-288. DOI= http://doi.org/10.1109/icaict.2015.7338564

[10] Afanasyev, A. N., Voit, N. N., and Kirillov, S. Y. 2017. Development of RYT-grammar for analysis and control dynamic workflows. In *International Conference on Computing Networking and Informatics* (Lagos, Nigeria, October 29-31, 2017). IEEE, 1-4. DOI= http://doi.org/10.1109/iccni.2017.8123797

[11] Afanasyev, A. N., Ukhanova, M. E., Ionova, I. S., and Voit, N. N. 2018. Processing of Design and Manufacturing Workflows in a Large Enterprise. In *Lecture Notes in Computer Science*, O. Gervasi et al., Eds. Springer, Cham, 565–576. DOI= http://doi.org/10.1007/978-3-319-95171-3_44

[12] Afanasyev, A. N., Voit, N. N., Gaynullin, R. F. 2016. The Analysis of Diagrammatic of Workflows in Design of the Automated Systems. In *Proceedings of the 12th International FLINS Conference (FLINS 2016)*, Roubaix, France, 509-514. DOI= http://doi.org/10.1142/9789813146976_0082

[13] Afanasyev, A. N., Voit, N. N., Timofeeva, O. G., and Epifanov, V. V. 2017. Analysis and Control of Hybrid Diagrammatical Workflows. In *Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17)*, A. Abraham et al., Eds. Springer, Cham, 124–133 (2017). DOI= http://doi.org/10.1007/978-3-319-68321-8_13

[14] Voit, N. N. 2017. Development of timed RT-grammars for analysis of business process at manufacturing and in cyber-physical systems. In *International Conference on Computing Networking and Informatics* (Lagos, Nigeria, October 29-31, 2017). IEEE, 1-5. DOI= https://doi.org/10.1109/iccni.2017.8123798