Development of RVT-grammar for analysis and control dynamic workflows

A.N. Afanasyev, N.N. Voit, S.Y. Kirillov
«Computing technique» department
Ulyanovsk State Technical University
Ulyanovsk, Russia
{a.afanasev, n.voit}@ulstu.ru, xayc73@gmail.com

***Abstract*—Authors develop RVT-grammar to analyze and control dynamic workflows. We consider aspects of dynamic workflows: ensemble (orchestration, choreography) and transformation with saving connections.**

***Keywords*—timed grammars; temporal logic; business process**

I. Introduction

The efficiency of business process can be improved with help dynamical reconfiguration of business enterprise process as workflows. There are some problems as deadlock, timing, safety and others where modification of the workflows using design, analysis, checking, modeling and transformation of dynamic workflows [1].

The design and development of automated systems must include the adaptation for agile requirements of an environment. In work [2], the agile is a main property of the manufacture to design. There are two behaviors of the enterprise. First of them is to change own business process for the existing manufacturing. Second of them is to create new marketable products. In order enterprise change own business process for increasing of product quality and to get new market outlets often. The work [3] presents that the speed of development in industry and technology need to change monolithic approaches.

A lot of the large enterprise as IBM, ARIS note that the monolithic product lifecycle management systems with static workflows are limited and not enough to successfully complete the product lifecycle [4]. There are not enough resources in approaches, automatic tools and the standardization of these management systems. Therefore, the large enterprise and companies have poorly designed workflows that stimulate increasing a cost of a product, and the improvement of product lifecycle becomes expense.

We used a definition given in [5] for a dynamic workflow as a process of adaptation to the current environment. ProBis [6] has monolithic workflows. Dynamic workflows are presented in works [7-9] with YAWL (Yet Another Workflow Language) and iPB. Also, we define the novel RVT-grammar as a timed (temporal) finite state grammar using a memory as stacks and elastic tapes.

There is an overview of works and the problem in Introduction. Orchestration and choreography as Ensemble are suggested in a section two. The novel grammar is presented in section three, and control diagrams as transformation is offered in a section four. In the end of this work, authors give the conclusion and future works.

II. The Ensemble

Orchestration is the inner workflows in an enterprise or a company that presents their inner business process [11]. IBM, Microsoft, Oracle, BEA Systems create tools as BPEL4WS, XLANG, WSFL to describe the business logic [12]. In order there is only one manager for orchestration to control these workflows in business process of enterprise.

Choreography is the outer workflows in a lot of enterprises and companies that have relationship each with other. Each member of choreography can describe a role and his place in workflows. All choreography relationships are monitored in log (Fig. 1).

A dynamic of workflows represented at two aspects: orchestration and exemplar of choreography which to add up to ensemble. An emerged exemplar of choreography to be in link with designed diagrams.

Organization 1 contains BPMN, IDEF0, UML AD, eEPC, UML class diagram in orchestration. BPNM, IDEF0, UML AD are in ensemble only [13, 14]. Organization 2 contains Java, C#, Timed automaton, Petri nets in orchestration, but Java and Timed automaton are in ensemble only.

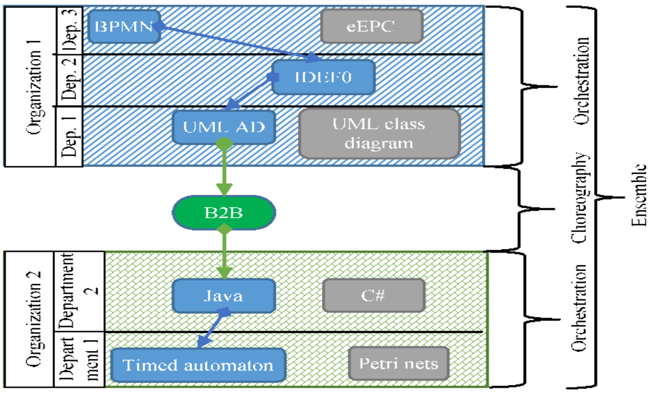

Fig. 1. Ensemble of divers diagrams and hybrid orchestration

978-1-5090-4642-3/17/$31.00 ©2017 IEEE

Let’s consider an orchestration of IDEF0, UML AD diagrams in ensemble (Fig. 1). IDEF0 diagrams performed very well in functional tools with links like “Input”, “Output”, “Control”, “Mechanism”. UML AD diagrams performed very well in developing tools of algorithm. In case when it is not necessary for considering control and mechanism at IDEF0 but it is necessary for considering tracking flows of process need to build UML AD diagram.

III. TIMED RVT-GRAMMAR WITH MEMORY

The authors have developed automaton grammar with memory, called RVT-grammar, to analysis and control diagrams with time for the agile management systems [15-23]. RVT-grammar can be written as:

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

where $V = \{v_l, l = \overline{1, L}\}$ is a set of helper alphabet, that has operations with the stack and the elastic tape;

$\Sigma = \{a_l, l = \overline{1, T}\}$ is a set of terminal alphabet, that has a group of primitive graphic objects and links;

$\tilde{\Sigma} = \{\tilde{a}_l, t = \overline{1, \tilde{T}}\}$ is a set of quasi-terminal alphabet, adding the terminal alphabet, that includes following quasi-terms: graphic objects; graphic objects that has more than one input; links with the specific semantic; the end of the analysis;

$R = \{r_i, i = \overline{1, I}\}$ is the production of this grammar, that has rules, and each complex r_i has P_{ij} of rules, where $r_i = \{P_{ij}, j = \overline{1, J}\}$;

$r_0 \in R$ is an axiom and a first complex, $r_k \in R$ is a last complex.

This production, including a set of rules, $P_{ij} \in r_i$ that is given as:

$$a_l^{[t_i]} \xrightarrow{W\nu(\gamma_1, \dots, \gamma_n)} r_m, \tag{2}$$

where $\mathrm{W}_\nu(\gamma_1, \dots, \gamma_\mathrm{n})$ is n-ary relation that has an operation with the stack and the elastic tape, depending on $\nu \in \{1, 2, 3\}$ (accordingly: 1 - read, 2 - write, 3 - compare); $r_m \in R$ is the receiver of rules;

C is a finite set of clock identifiers;

$E -$ is a set of time expressions C (clock limitation and clock reset), is limited by the following expressions: from the beginning $\{c := 0\}$ and onwards $\{c \sim x\}$, and c is a variable, and x is a constant, $\sim \in \{=, <, \leq, >, \geq\}$;

$T \in \{t_1, t_2, t_3 \dots, t_n\}$ is a set of timed labels with given functions $FT\Sigma: \Sigma \times C \times E \to T$ and $FT\tilde{\Sigma}: \tilde{\Sigma} \times C \times E \to T$ accordingly.

The memories are stacks and elastic tapes. Stacks have an information about processing graphic objects with two or more outputs as a quasi-term. Elastic tapes have an information about processing graphic objects with two or more inputs as numbers. The writings, reading and comparing of data in these elastic tapes performs with help cells that defined on natural numbers.

The chain of $\varphi = \alpha_{l1}, \alpha_{l2}, \dots, \alpha_{l\lambda}$ is called RVT-derivation $\alpha_{l\lambda}$ from α_{l1} and it is denoted $\alpha_{l1} \overset{RV}{\Rightarrow} \alpha_{l\lambda}$ if for any $\xi < \lambda$ and $r_e \in R$ are as $\alpha_{l\xi+1} \in r_e$,$\left(a_l^{[t_i]} \xrightarrow{W\gamma(\gamma_1, \dots, \gamma_n)} r_e\right) \in r_i$.

The analysis of this chain begins from the first terminal with help a rule r_0. Step by step we use rules r_i for the analysis of this chain. There is a rule r_k in the end of the analysis. Operations with memory of this RVT-grammar perform when reading, writing and comparing cells of this chain. So, the stack is empty in the start and in the end of this analysis also. The elastic tape has a cortege of natural numbers in the end of the analysis.

IV. THE TRANSFORMATION

Dynamic reconfiguration of business process need to have a mechanism for transformation of diagrams reaching flexibility, improving a functional and an efficiency of enterprise’s business process. In work [24-26] the problem of reconfiguration has been researched both theoretical and practical.

Authors offer applying the structure transformation of a diagram with help procedures: delete, insert and replace with saving a connection during an interval of time.

It is necessary all graphic element have a timed label where we can define time of the transformation. As rule BPMN, eEPC, IDEF0, UML AD etc. graphic elements contain a description (notes in UML AD) which can be define as a timed variable.

Let’s see an example of UML AD diagram (Fig. 2).

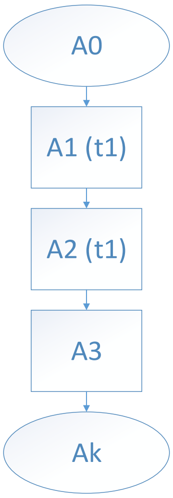

Fig. 2. UML AD diagram with t1 timed label

Graphic elements A1 and A2 have t1 timed labels. This means that a current element will be transform at t1 time with help operations: (1) Insert, (2) Replace, (3) Delete.

Reasoning to suppose that only one operation cab be performed at one element. Therefore, timed label is assigned to a tape where an element has their variants: number 1 – Insert, number 2 – Replace, number 3 – Delete.

Additional information when Insert and Replace saved at extended tape allowing to save both numbers and quasi-terms.

Additional Insert() function is used for the operation 1 allowing to get needed information from extended tape and form inserted fragment.

Operation 2 is a complex operation that represents an aggregate of removing and inserting operations. Replace() additional function is brought for ease.

Deleting is considered in a start. The diagram has a form in t1 time (Fig. 3).

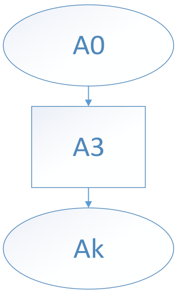

Fig. 3. Deleting elements at diagram

The chain including deleting element can be infinite size. Authors suggest the approach to perform deleting. If we meet element with timed label, then timed label is put in a stack. Next step an automaton follows about elements while not getting element with absent timed label. In this case it perform change_rel() special function that pop up from the stack timed label at deleting element and assign its with a current element. This algorithm is shown in Fig. 4.

In order not to leave deleting elements suspended in a diagram when to pass deleting quasi-term delete () function perform that delete elements from the diagram. delete_with_link() function performs deleting elements with an enter link.

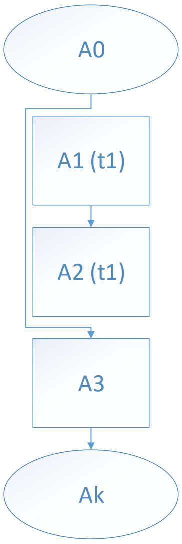

Fig. 4. Assignment of links where deleting an element

The grammar for that diagram is shown in Table 1.

TABLE I. TIMED RVT-GRAMMAR FOR UML AD

Prev. state	Quazi-term	Next state	Operation
r0	A0i	r1	insert()/W3(kt(1)==1)
	A0	r1	o
r1	rel	r2	o
r2	Ai	r1	insert()/W3(kt(1)==1)
	Ar	r1	replace()/W3(kt(1)==2)
	Ad	r3	(delete(), W1(l1m))/ W3(kt(1)==3)
	A	r1	o
	Ak	r5	o
r3	drel	r4	o
r4	Ai	r1	(change_rel(),insert())/W3(kt(1)==1)
	Ar	r1	(change_rel(), replace())/W3(kt(1)==2)
	Ad	r3	delete_with_link()/W3(kt(1)==3)
	A	r1	change_rel()
	Ak	r5	change_rel()
r5	no_label	rk	*

V. CONCLUSION

There is the problem with workflows for checking and exchanging the different formats between the large industrial enterprise, and also between their department. Time begins to play the main meaning for manufacturing that uses Internet of things often. This is why authors develop timed RVT-grammar to analyze, control and transform dynamic design workflows where time is a famous role. We create this grammar as a temporal finite state grammar that uses a memory like stack. So, this grammar lets to remove several semantic errors (structural-behavioral) at a stage of conceptual design in complex computer systems, and also to solve reengineering tasks for the reactive systems using real-time. As well as the transformation can check bad elements at a diagram, and make a refurbishment of this diagram. We show a simple example using AD UML diagrams. We will define a set of typical structural-behavioral errors with help to research for books and works, and try to extend it with own checklist in future works.

ACKNOWLEDGMENT

This research is supported by the grant of the Ministry of Education and Science of the Russian Federation, the project № 2.1615.2017/4.6. The reported study was funded by RFBR and Government of Ulyanovsk Region according to the research project № 16-47-732152. The reported study was funded by RFBR according to the research project № 17-07-01417.

REFERENCES

[1] J. C. P. Aguilar, K. Hasebe, M. Mazzara, and K. Kato, “Model Checking of BPMN Models for Reconfigurable Workflows”. URL: https://www.researchgate.net/publication/304788360_Model_Checking_

of_BPMN_Models_for_Reconfigurable_Workflows Cited October 18, 2017.

[2] B. Sherehiy, W. Karwowski, J. K. Layer, “A review of enterprise agility: Concepts, frameworks, and attributes,” International Journal of Industrial Ergonomics, vol. 37, pp.445-460, May 2007.

[3] J. Highsmith, K. Orr, A. Cockburn, “Extreme programming,” E-Business Application Delivery, pp. 4-17, Feb. 2000.

[4] A global Swiss company offering advanced intelligent application software for multiple business sectors, 2016. URL: http://whitestein.com/ Cited October 18, 2017.

[5] I. Bider, A. Jalali, “Agile Business Process Development: Why, How and When Applying Nonaka's theory of knowledge transformation to business process development. Information Systems and e-Business Management,” Springer-Verlag Berlin Heidelberg, 2014. URL: https://www.researchgate.net/publication/266078141 Cited October 18, 2017.

[6] T. Andersson, A. Andersson-Ceder, and I. Bider, “State flow as a way of analyzing business processes-case studies,” Logistics Information Management, vol. 15(1), pp. 34-45, 2002. URL: http://www.ibissoft.com/publications/Cases.pdf Cited October 10, 2017.

[7] YAWL Foundation, YAWL. URL: http://www.yawlfoundation.org/ Cited October 18, 2017.

[8] I. Bider, “Analysis of Agile Software Development from the Knowledge Transformation Perspective,” in Perspectives in Business Informatics Research. BIR 2014. Lecture Notes in Business Information Processing, vol 194, Johansson B., Andersson B., Holmberg N., Eds. Springer, Cham, 2014.

[9] IbisSoft, “iPB Reference Manual,”. URL: http://docs.ibissoft.se/node/3 Cited October 18, 2017.

[10] T. Bultan “Analyzing Interactions of Asynchronously Communicating Software Components,”. In: Beyer D., Boreale M. (eds) Formal Techniques for Distributed Systems. Lecture Notes in Computer Science, vol 7892. Springer, Berlin, Heidelberg, 2013. URL: https://link.springer.com/chapter/10.1007/978-3-642-38592-6_1 Cited October 18, 2017.

[11] Orchestration and Workflow. URL: https://www.cloudenablers.com/blog/orchestration-and-workflow/ Cited October 18, 2017.

[12] W. M. P. Van der Aalst, “Don’t go with the flow: Web services composition standards exposed,” IEEE intelligent systems, vol. 18, no. 1, pp. 72-76, 2003. URL: http://www.martinfowler.workflowpatterns.com/documentation/documents/ieeewebflow.pdf Cited October 18, 2017.

[13] D. A. Marca and C. L. McGowan. SADT: structured analysis and design technique. McGraw-Hill, Inc., 1987. URL: http://dl.acm.org/citation.cfm?id=31837 Cited October 18, 2017.

[14] TP026B, Rev. "Rational Unified Process.". URL: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf Cited October 18, 2017.

[15] A. Afanasyev, N. Voit, “Intelligent Agent System to Analysis Manufacturing Process Models,” Proceedings of the First International Scientific Conference «Intelligent Information Technologies for Industry» (IITI’16) vol.451 of the series Advances in Intelligent Systems and Computing. Russia, 2016, pp. 395-403.

[16] A. Afanasyev, N. Voit, R. Gaynullin, “The Analysis of Diagrammatic Models of Workflows in Design of the Complex Automated Systems,” Proceedings of the First International Scientific Conference «Intelligent Information Technologies for Industry» (IITI’16) vol. 450 of the series Advances in Intelligent Systems and Computing. Russia, 2016, pp. 227-236.

[17] A.N. Afanasyev, N.N. Voit, R.F. Gainullin, “Diagrammatic models processing in designing the complex automated systems,” 10th IEEE International Conference on Application of Information and Communication Technologies (AICT). Baku, Azerbaijan, 2016, pp. 441-445.

[18] A. Afanasyev and N. Voit, “Multi-agent system to analyse manufacturing process models,” Proceedings of International conference on Fuzzy Logic and Intelligent Technologies in Nuclear Science - FLINS2016. France, 2016, pp. 444-449.

[19] A.N. Afanasyev, N.N. Voit, E.Yu. Voevodin, R.F. Gainullin, “Control of UML diagrams in designing automated systems software,” Proceedings of the 9th IEEE International conference on Application of Information and Communication Technologies: AICT – 2015, 2015, pp. 285-288.

[20] O.G. Sharov, A.N. Afanas’ev, “Syntax-directed implementation of visual languages based on automaton graphical grammars,” Programming and Computer Software, vol. 6. pp. 56-66, 2005.

[21] O.G. Sharov, A. N. Afanas’ev, “Neutralization of syntax errors in the graphic languages,” Programming and Computer Software. 2008, vol. 1, pp. 61-66, 2008.

[22] O.G. Sharov, A. N. Afanas’ev, “Methods and tools for translation of graphical diagrams,” Programming and Computer Software. 2011. vol. 3, pp. 65-76, 2011.

[23] A.N. Afanasev, N.N. Voit, E.Yu. Voevodin, R.F. Gainullin, “Analysis of Diagrammatic Models in the Design of Automated Software Systems,” Object Systems – 2015: Proceedings of X International Theoretical and Practical Conference (Rostov-on-Don, 10-12 May, 2015) / Edited by Pavel P. Oleynik. – Russia, Rostov-on-Don: SI (b) SRSPU (NPI), 2015, pp. 124-129.

[24] M. Mazzara, N. Dragoni, M. Zhou, “Dependable Workflow Reconfiguration in WS-BPEL,” 5th Nordic Workshop on Dependability and Security (NODES, 2011), 2011. URL: http://deploy-eprints.ecs.soton.ac.uk/310/ Cited October 18, 2017.

[25] M. Mazzara, F. Abouzaid, N. Dragoni, A. Bhattacharyya, “Toward Design, Modelling and analysis of Dynamic Workflow Reconfiguration,” A Process Algebra Perspective. 8th International Workshop on Web Services and Formal Method (WSFM 2011), 2011. URL: http://orbit.dtu.dk/fedora/objects/orbit:71336/datastreams/file_6251396/content Cited October 18, 2017.

[26] F. Abouzaid, A. Bhattacharyya, N. Dragoni, J. S. Fitzgerald, M. Mazzara and M. Zhou, “A Case Study of Workflow Reconfiguration: Design, Modelling, Analysis and Implementation,” Newcastle University, Technical report CS-TR-1265, July 2011. URL: http://www.cs.ncl.ac.uk/publications/trs/papers/1265.pdf Cited October 18, 2017.