УДК 004

TIMED RV-GRAMMAR AND ONTOLOGY FOR CHECKING LEARNING BUSINESS PROCESS

A.N. Afanasyev[20], N.N. Voit[21]

Abstract: Authors offer timed grammar and ontology to check any notations that formed as business process using Extend EPC Notation and etc. Semantical errors are catch with this grammar. For instance, the learning process is presented by Extend EPC Notation. We check them on errors using our timed RV-grammar.

Keywords: RV-grammar; ontology; learning business process.

ВРЕМЕННАЯ RV-ГРАММАТИКА И ОНТОЛОГИЯ ДЛЯ ПРОВЕРКИ БИЗНЕС-ПРОЦЕССА ОБУЧЕНИЯ

А.Н. Афанасьев, Н.Н. Войт

Аннотация: Авторы предлагают временную грамматику и онтологию для проверки любых нотаций, которые формируются как бизнес-процесс посредством расширения нотаций EPC и др. Данная грамматика находит семантические ошибки. Например, процесс обучения представлен расширением нотаций EPC. Мы проверяем их на ошибки, используя нашу временную RV-грамматику.

Ключевые слова: RV-грамматика; онтология; бизнес-процесс обучения.

Introduction

Workflow is a trace for executing a variety of business process tasks, taking into account constraints and business events, containing time constraints and data flows. It is necessary to identify and correct errors in the processes in order to avoid failures. Although errors can occur in the cause-effect relationships between tasks, we focus on the semantic errors in the execution of the workflow, namely denotative and significative semantics. Denotative semantics determines the errors of antonymy, the synonymy of words in the business events of the workflow. Significative semantics reveals structural errors in the

[20] *Ulyanovsk, Ulyanovsk State Technical University, e-mail: a.afanasev@ulstu.ru.*
[21] *Ulyanovsk, Ulyanovsk State Technical University, e-mail: n.voit@ulstu.ru.*

workflow on the basis of isomorphism and trace homomorphism. Ad-hoc in the workflow is an add-on and makes the process not strict, thereby violating the canonical rules of the process. Performing such a workflow can lead to a decrease in customer satisfaction, an increase in employee overload, a decrease in brand image, a decrease in profits, and a significant expenditure of management time. Thus, identifying and eliminating semantic errors in workflows is important for business.

The work flow should be conceptually presented in the formal language for analysis and expertise before deployment in a real business environment. This view is also useful when transferring workflow tasks between designers, users, process engineers, managers and technical personnel. In addition, process models in the presentation can be tested using approaches that have a corresponding formal language to determine the workflow. Conceptual representations can be performed using Workflow Nets (WF-nets), Workflow Graphs, Object Coordination Nets (OCoNs), Adjacency Matrix, Unified Modeling Language (UML) diagrams, Evolution Workflow Approach and Propositional Logic. Currently, validation algorithms exist for WF-nets, Workflow Graphs, UML diagrams, Propositional Logic and Adjacency Matrix representations. And popular algorithms are those that are based on WF-nets and Workflow Graphs. WF-nets are based on Petri nets, and many formal methods for analyzing Petri nets were used to obtain theoretical solutions to the problems encountered in the design of WF networks. Although many complex language constructs that are useful in a business environment can be implemented using WF-nets, the Workflow Management Council (WfMC) uses only six basic language of the process language. WfMC adopted this approach to keep the simulation very simple and clear.

For a business event, a subset of workflow tasks is performed in accordance with the object data (customer data, environment data, business process data, and business domain data), for example, such as ordering. This subset of tasks, together with the workflow used to execute the business process, is called an instance. Until now, most workflow management systems (WfMSs) provide only modeling tools for testing workflow models using the trial and error method [8-10]. These modeling tools can be used to perform a subset of workflow instances to check for structural conflicts that may occur in the respective scenarios. However, workflows can have many instances, and the verification task becomes difficult for all instances.

Checking the workflows for structural and semantic errors is a computational task, so different formal approaches and languages can be used for this. However, the approach taken for verification should support the language of the workflow description. Because of the computational complexity of the problem (polynomial, exponential), only a few approaches successfully cope with the verification of workflows, taking into account constraints, including temporary ones, for all types of workflow graphs.

Article has the following structure. In Introduction, the list of standard problems with workflows is submitted briefly. Related work has an overview of works on this topic. In Timed RV-grammar, Timed automaton, Ontology and List of errors, authors describe the approach. In Example, the implementation suggested approach is presented. Outputs and the further directions of researches are presented in the conclusion.

1. Timed RV-grammar

Timed RV-grammar is defined as the tuple [1-7]

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

where $V = \{v_e,\ e = \overline{1, L}\}$ is an additional alphabetic for the operation onto a memory; $\Sigma = \{(a_l, t_l), l = \overline{1, T}\}$ is an alphabetic of events; $\tilde{\Sigma} = \{\tilde{a}_n,\ n = \overline{1, \tilde{T}}\}$ is a quasi-term alphabet, extending Σ; C is a time identifier or clock; E is a set of the temporal relations as $\{c \sim x\}$, where c is a variable (a time identifier or clock), $c = c + t_{l-1}$, and x is a constant (timestamp or t_l), $\sim \in \{=, <>, <, \leq, >, \geq\}$; $R = \{r_i,\ i = \overline{0, I}\}$ is a rule of this grammar G (a set of production rule's complexes), where this complex r_i has a subset P_{ij} of the production rule $r_i = \{P_{ij},\ j = \overline{1, J}\}$; $\tau = \{t_l, l = \overline{1, T}\}$ is a set of timestamps, where $E \subseteq C \times \tau$; $r_0 \in R$ is an axiom of this grammar (a name of the first production rule), $r_k \in R$ is the last production rule. The production rule $P_{ij} \in r_i$ has a view as

$$(a_l, t_l) \xrightarrow[E]{W_\gamma(v_1, \dots, v_n)} r_m. \quad (2)$$

where $W_\gamma(v_1, \dots, v_n)$ is n-relation, that defines a type of an operation over memory, depending on $\gamma = \{1,2,3\}$ (1 – write, 2 – read, 3 – compare); (a_l, t_l) is a word as a pair of an event and a timestamp; $r_m \in R$ is a name of a target production rule. The language $L(G)$ of this grammar has words as (a_l, t_l) and presents a trace $\sigma = \{a_0\} \to \{a_l, t_l\} \to \{a_k\}$.

2. Timed automaton

The timed automaton *TimedA* isrepresented by following components:

$$TimedA = (V, \sum, C, E, \delta, S_0, S, S_k), \quad (3)$$

where $V = \{v_e, \ e = \overline{1,H}\}$ is an auxiliary alphabet (the alphabet of operations over internal memory); $\sum = \{(a_l, t_l), \ l = \overline{1,L}\}$ is a terminal alphabet of a language; 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, $\sim \in \{=, <, \leq, >, \geq\}$; $S = \{S_i, \ i = \overline{0,I}\}$ is a set of states; $S_0 \in S$ is a set of a beginning state; $\delta: S_i \times (a_l, t_l) \xrightarrow[E]{W_\gamma(v_1,\ldots,v_n)} S_m$ is the ratio of transitions, where $W_\gamma(v_1, \ \ldots \ , \ v_n)$ is a n-th relation, which determines the type of operation over the internal memory depending on $\gamma \in \{0,1,2,3\}$ (respectively, 0 – operation is not performed, 1 – record, 2 – read, 3 – compare); $r_i \in R$ is the name of the complex of a production rule's source; $S_m \in S$ is the name of the state of a production rule's successor.

3. Ontology

The ontology is represented as follows:

$$O = (Class, Property, Relation, Axiom). \quad (4)$$

where $Class$ is the set of concepts (classes) defined for a particular subject area; $Property$ is the set of properties of a concept; $Relation$ is the set of semantic links defined between concepts in $Class$. Many types of relationships are the following: one to one, one to many and many to many. The set of basic relations are: (synonymy), (a kind of something), f (part of something), (instance of something), property_of (property of something); $Axiom$ is the set of axioms. An axiom is a real fact or a rule that determines the cause-effect relationship.

4. List or errors

We can catch the following errors [8-10]:

1. Thecyclicconnection;
2. Mutuallyexclusivelinks;
3. Multiplecommunication;
4. Remotecontexterror;
5. Control transfer failure;
6. Error in the multiplicity of inputs;
7. Errormultiplicityofoutputs;
8. Invalid link;
9. Communication error;
10. Access level error;
11. Error transmitting the message;
12. An error in the delegation of control;

13. A quantitative error in the elements of the diagram;
14. Excluding links of the wrong type;
15. A call directed to the life line;
16. Collapsed connection;
17. Violation of the multiplicity of dependencies;
18. Mutually exclusive links;
19. Synchronous call before receiving a response;
20. Great synonymy;
21. The antonymy of objects;
22. Conversion of relations;
23. Inconsistencyofobjects.

5. Example

On Figure 1 we can see a business process of testing.

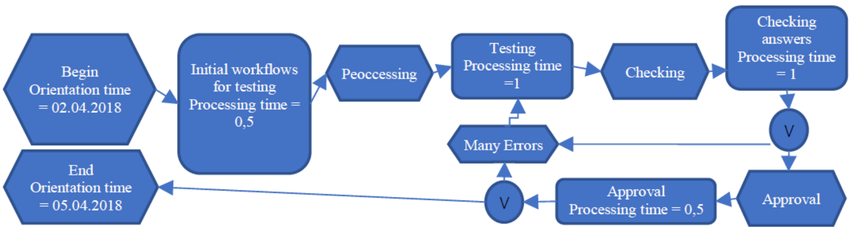

Figure. 1. The diagram for testing

The one automated store m and the one tape $1^t$are used by this grammar as the internal memory. The timestamp tm is recorded on a tape. In Table 1, we can see the written Timed RV-grammar for the Figure 1.

Table 2. Timed RV-grammar for the example 1

Sourceproduction rule'scomplex	Quasi-term	Targetproduction rule'scomplex	Relation
r_0	B	r_1	$W_1(1^{1m})$, $W_1(tm^{1t})$
r_1	W	r_2	$W_1(2^{1m})/W_3(c<=tm^{1t})$
r_2	C	r_3	$W_1(3^{1m})/W_3(c>tm^{1t})$
r_2	C	r_4	$W_3(c<=tm_{1t})$
r_3	I	r_4	c=0, $W_2(3^{1m})/W_3(c<=tm^{1t})$
r_4	A	r_3	$W_2(3^{1m})/W_3(c>tm^{1t})$
r_4	A	r_k	$W_2(2^{1m})$, $W_2(1^{1m})$
r_k	E	–	–

Let's write a timed automaton for Timed RV-grammar. The alphabet of an event process is a set of

$\sum = \{Begin, Processing, Checking, ManyErrors, Approval, End\}$.

$S = \{B, W, C, I, A, E\}$. Let's define δ (Table 1).

Table 1. A matrix of δ

Constraint	B	W	C	I	A	E
B		$W_1(1^{1m})$, $W_1(tm^{1t})$				
W			$W_1(2^{1m})/$ $W_3(c<=tm^{1t})$			
C				$W_1(3^{1m})/$ $W_3(c>tm^{1t})$	$W_3(c<=tm^{1t})$	
I					$c=0$, $W_2(3^{1m})/$ $W_3(c<=tm^{1t})$	
A				$W_1(3^{1m})/$ $W_3(c>tm^{1t})$		$W_2(2^{1m})$, $W_2(1^{1m})$
E						

Let's depict an ontology for this timed automaton. We transform the automaton into an ontology, replacing the *States* with *Classes*, adding properties to the notions (*Property*). We get a graphical representation of the ontology with class properties (Figure 2).

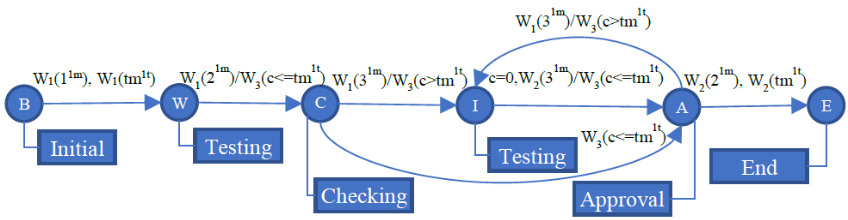

Figure 2. The graph of the timed automaton for the Figure 1

We can see twice Testing in a circle W and in a circle I. So, this diagram has an error as Great synonymy.

Conclusion

We have developed an approach to analyzing workflows of business processes for errors according to the list of errors. Checking not only structural errors, but also semantic errors, distinguishes work from existing ones. The temporary automaton grammar offered by the authors has a linear characteristic of the workflow analysis time, takes into account the language of the process description and can be applied to any diagram. Also, the time machine allows simulation of the process in visual form. The ontological model is the main one for analysis for denotative and significative errors in workflows. The authors finalized the list of structural and semantic errors encountered in workflows. In future works, the authors want to increase the number of examples of application of the approach in industry, training, cyber-physical systems, in the development of automated systems.

Acknowledgments

The reported study was funded by RFBR according to the research project № 17-07-01417.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.

References

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

2. Afanasyev, A., Voit, N.: Intelligent Agent System to Analysis Manufacturing Process Models. First International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'16), vol. 451 of the series Advancesin Intelligent Systems and Computing. Russia, pp. 395-403 (2016).

3. Afanasyev, A.N., Voit, N.N., Voevodin, E.Yu., Gainullin, R.F.: Controlof UML diagrams in designing automated systems software. 9th IEEE International conference on Application of Information and Communication Technologies: AICT-2015, pp. 285-288 (2015).

4. Afanasyev, A., Voit, N.: Multi-agent system to analyse manufacturing process models. International conference on Fuzzy Logic and Intelligent Technologies in Nuclear Science - FLINS2016. France, pp. 444-449 (2016).

5. Afanasyev, A., Voit, N., Gainullin, R.: The analysis of diagrammatic models of workflows in design of the complex automated systems. International conference on Fuzzy Logic and Intelligent Technologies in Nuclear Science - FLINS2016. – France, pp. 509-514 (2016).

6. Afanasyev, A. N., Voit, N.N., Kirillov, S. Y.: Development of RYT-grammar for analysis and control dynamic workflows. 2017 International Conference on Computing Networking and Informatics (ICCNI) 2017, 1-4 (2017), doi: 10.1109/ICCNI.2017.8123797, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8123797&isnumber=8123766, last accessed 2017/03/15.

7. Voit, N. N.: Development of timed RT-grammars for analysis of business process at manufacturing and in cyber-physical systems. 2017 International Conference on Computing Networking and Informatics (ICCNI) 2017, 1-5 (2017), doi: 10.1109/ICCNI.2017.8123798,

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8123798&isnumber=812376, last accessed 2017/03/15.

8. Y. Wang, Y. Fan, Using Temporal Logics for Modeling and Analysis of Workflows, in Proc. of E-Commerce Technology for Dynamic E-Business, IEEE International Conference on. doi: 10.1109/CEC-EAST.2004.72 (2004).

9. N. Saeedloei and G. Gupta, Timed definite clause ω-grammars, In Proc. of/ Technical Communications of the International Conference on Logic Programming (2010). URL: http://www.floc-conference.org/ICLP-home.html

10. L. Fischer, Workflow Handbook (2005).