Temporal automaton mechanism for transforming hybrid dynamic design workflows diagrammatic models*

N. N. Voit†, S. Yu. Kirillov, S. I. Bochkov, M. E. Ukhanova and I. S. Ionova

"Computing Technique" department, Ulyanovsk State Technical University, Ulyanovsk, 432027, Russia

†E-mail: n.voit@ulstu.ru

www.ulstu.ru

The authors have developed a new method (mechanism) for transforming diagrammatic models in the basis of graphic languages. This method takes into account the syntax (topology), denotative and significative semantics of the transformation. Thanks to the method the execution time of design workflows during the CAD systems design is reduced, and the workflows quality is also improved.

Keywords: Worfklows; Diagrammatic language; Grammar.

1. Introduction

In the business process management theory, the paradigm of business processes (workflows) digitalization and handling of the end-to-end digital design in Industry 4.0 while automating complex technical systems design contains key analysis and synthesis design procedures. These procedures are connected with the latest research directions and significantly affect the design result. Moreover, the problem of the design decisions success in this theory has been dealt with for more than 30 years. Such attention to the problem is caused by a high degree of development (design decisions) going beyond the planned time, financial and functional parameters. In the existing theory, the reasons are identified and recommendations are made to increase the complex computer-aided systems design success. However, according to the Standish Group [5] engaged in research in the field of successful development of automated systems, only 40% of developments are currently being completed successfully.

The workflows presented in the diagrammatic basis using the visual languages UML AD [10-11], IDEF [12], ER, DFD, eEPC, SDL, BPMN [14],

* The reported research was funded by Russian Foundation for Basic Research and the government of the region of the Russian Federation, grant № 18-47-730032

†

SharePoint, etc. play an important role in achieving success in developing complex automated systems. The use of such models increases both the design process efficiency and quality by unifying the means of interaction between customers and designers, rigorous documentation of design and architectural, functional solutions and formal control to correctness of diagrammatic notations.

The article contains a review of scientific research on this topic, the author's temporal automaton mechanism for transforming diagrammatic models of hybrid dynamic flows of design work, and also an example of transforming a diagrammatic model in the basis of the graphical language UML AD [10-11] is considered.

2. Related Works

Workflow analysis methods can be used to study the qualitative and quantitative characteristics of design work flows. Qualitative characteristics are understood as logical and algebraic correctness of workflows formalized using graph theory, workflow networks, matching matrices, graphic modeling languages, including Unified Model Language, Business Process Management Notation [14], IDEF0 and eEPC, etc., as well as the evolutionary approach, propositional logic, etc. [13]. Quantitative characteristics represent the efficiency of the execution of work flows by parameters, for example, such as the average time of maintenance, utilization of production capacities (simple equipment), etc. Evaluation of the effectiveness of workflows is carried out using simulation modeling (Petri nets), Markov chains and queuing theory (queuing systems), etc.

The model checking method has found quite wide application for workflow analysis in developing error-free systems at the stage of conceptual design. The model checking approach is intended for analysis, control of workflows by means of formal verification of whether a given logical formula is satisfied on a given structure (whether a given logical formula Φ is true for a given transition system M, i.e., whether M will be a model of Φ). The main drawback of the approach is the study of the model, not the system itself, so the question of the adequacy of the model to the system arises, while the complexity of solving the above problems is exponential. Model checking is intended for experienced scientists and engineers, as it is difficult to understand and use. Workflows are also specified by managers, designers, laboratory assistants, technologists who are not trained in the field of formal models and informatics, and for formal analysis a detailed representation of the process model in a formal language is necessary, which is difficult to construct and understand for them. The model checking approach is actively researched by Karpov [6].

In deductive verification, the workflow is verified, which reduces to proving theorems in a suitable logical system using axioms and inference rules (for example, using the Prolog language, automaton grammars, etc.). This highly complex procedure cannot be fully automated; it requires the participation of a person acting on the basis of assumptions and guesswork, using intuition to construct invariants and a non-trivial choice of alternatives.

For example, the deductive verification method (prepositional logic) and its application are given in the work of Henry H. Bi and J. Leon Zhao [7]. The complexity of this method is $O(N^2)$, where N is the tasks count in the workflow. This method is not complete, since it cannot detect structural errors in all types of overlapping workflow structures.

Saeedloei and Gupta [8] used a temporal automaton that implements a temporal context-free grammar to analyze cyber-physical systems and then translate this grammar into a program for the Prolog interpreter.

Currently, π-calculus is a promising, but still very young and developing theory, it has many open questions and unsolved problems.

Wang and Fan [9] propose using temporal logic of actions to describe workflows in graph form, which requires a description of all graph routes in the formulas of temporal logic of actions. In this case, linear temporal logic is applied to formalize a route from tasks, AND, OR branches, and JOIN convergences, however, the question of the adequacy of constructing a description of workflows in graph form remains unresolved.

3. Temporal machine transformation mechanism

The diagrammatic models transforming mechanism is based on their dynamic reconfiguration based on RVTI-grammar [15] in order to achieve flexibility, improve functionality and increase the efficiency of the existing enterprise business process. In [1-4], the reconfiguration problem was deeply studied from both theoretical and practical points of view. The authors propose to apply the transformation of the diagram structure using the delete, insert and replace procedures while maintaining connectivity for (before, after, etc.) a specific time using the RVTI-grammar. To do this, it is necessary that all graphic primitives have a time stamp, which determines the time of transformation of the chart. Typically in BPMN, eEPC, IDEF0, UML AD, etc. graphic primitives contain a description (note in UML AD) that can be defined as a time variable.

Consider an example of a UML AD diagram (Figure 1).

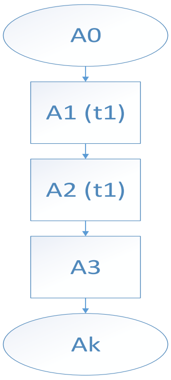

Fig. 1. UML AD diagram with timestamp t_1

Graphic primitives (elements) A_1 and A_2 have timestamps t_1. This means that at a certain point in time t_1, certain transformations (operation) will be performed with these elements: (1) Insert, (2) Replace, and (3) Delete. It is logical to assume that only one operation can be performed on one element at a time. Therefore, for each time stamp, a ribbon will be allocated on which for one element it will be possible to indicate three options: 1 - Insert, 2 - Replace, 3 - Delete. Additional information during the Insert / Replace operation will be stored in an extended tape, allowing to store not just numbers, but many quasitherms. For operation 1, we will use the additional insert() function, which allows us to extract the necessary information from the expanded tape and form the inserted fragment due to the sub-grammar. Operation 2 is a complex one and is a collection of delete and insert operations. For this, an additional replace() function is introduced. At the initial stage, deletion is considered. As a result, at time t_1, the diagram takes the form shown in Figure 2.

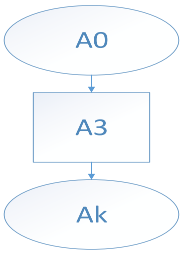

Fig. 2. Removing items in a chart

The chain of deleted items can be arbitrarily long. To perform the removal, we will use the following method. If an item with a marked time stamp is found, then a link to this item is pushed into the stack. Next, the machine follows the elements until it encounters an element with no timestamp. In this case, the special *change_rel()* function is executed, which pops the link to the initially deleted item from the stack and binds it to the current item. The process is shown in Figure 3.

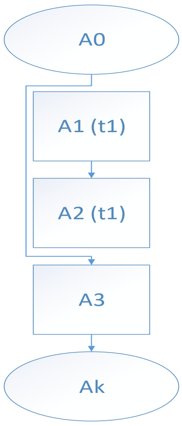

Fig. 3. Reassigning relationships when deleting an item

In order not to leave deleted elements hanging on the diagram when passing through deleted quasitherms, the *delete()* function is executed, which removes the element from the diagram. Also the *delete_with_link()* function exists which will remove the item along with the incoming link.

The RVTI-grammar for converting such a diagram is presented in Table 1.

Table 1. Temporal RVTI-grammar [15] for UML AD

Previous state	Quasitherm	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	*

4. Conclusions

A new temporal automaton mechanism for converting diagrammatic models of hybrid dynamic design workflows has been developed and investigated. This mechanism takes into account syntax (topology), denotative and significative semantics through the use of author RVTI-grammar. It provides a reduction in the design workflows execution time in the large industrial CAD systems design. The mechanism also improves the quality of these diagrammatic models by taking into account new revealed types of semantic errors (due to the use of RVTI-grammar).

Further research is related to the interpretation of such workflows, the synthesis of diagrammatic models based on the “design footprints” of the designer in the basis of graphic languages.

Acknowledgements

The reported research was funded by Russian Foundation for Basic Research and the government of the region of the Russian Federation, grant № 18-47-730032.

References

1. A. N. Afanasyev and N. N. Voit, Intelligent Agent System to Analysis Manufacturing Process Models, in *Proceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry”(IITI’16)*, (Rostov-on-Don, Russia, 2016).
2. A. N. Afanasyev, N. N. Voit and R. F. Gaynullin, The Analysis of Diagrammatic Models of Workflows in Design of the Complex Automated Systems, In *Proceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry”(IITI’16)*, (Rostov-on-Don, Russia, 2016).
3. A. N. Afanasyev, N. N. Voit and R. F. Gaynullin, Diagrammatic models processing in designing the complex automated systems, In *IEEE 10th International Conference on Application of Information and Communication Technologies (AICT’16)*, (Baku, Azerbaijan, 2016).
4. A. N. Afanasyev et al. Control of UML diagrams in designing automated systems software, In *IEEE 9th International Conference on Application of Information and Communication Technologies (AICT’ 15)*, (Rostov-on-Don, Russia, 2015).
5. Standish Group. https://www.standishgroup.com/

6. Yu. G. Karpov, *MODEL CHECKING. Verification of concurrent and distributed software systems* (BHV-Petersburg, Saint-Petersburg, 2010).
7. H. Bi Henry and J. Leon Zhao, Applying Propositional Logic to Workflow Verification, *Information Technology and Management* **5**(3-4), 293–318 (2004).
8. N. Saeedloei and G. Gupta. Timed definite clause omega-grammars, in *Technical Communications of the 26th International Conference on Logic Programming*, eds. M. Hermenegildo and T. Schaub (Dagstuhl, Germany, 2010), pp. 212-221.
9. Y. Wang and Y. Fan. Using Temporal Logics for Modeling and Analysis of Workflows, in *IEEE Proceedings of E-Commerce Technology for Dynamic E-Business*, (Beijing, China, 2004) doi: 10.1109/CEC-EAST.2004.72
10. G. Booch, I. Jacobson and J. Rumbaugh, *The Unified Modeling Language User Guide* (Addison-Wesley, 1998).
11. P. Kruchten, *The Rational Unified Process* (Addison-Wesley, 1998).
12. D.A. Marca and C.L. McGowan, *SADT: Structured Analysis and Design Techniques* (McGraw-Hill, 1988).
13. G. Booch, *Object-oriented Analysis and Design with Applications, 2nd edition* (Addison-Wesley, 1994).
14. BPMN Specification - Business Process Model and Notation. http://bpmn.org
15. A. N. Afanasyev, N. N. Voit and S. Yu. Kirillov. Development of RYT-grammar for analysis and control dynamic workflows, in *International Conference on Computing Networking and Informatics (ICCNI'17)*, (Lagos, Nigeria, 2017).