Processing of Diagrammatic Workflow on the Basis of Finite State Grammars

© Alexander Afanasyev © Nikolay Voit © Sergey Kirillov © Maria Ukhanova © Irina Ionova
Ulyanovsk State Technical University,
Ulyanovsk, Russia
a.afanasev@ulstu.ru n.voit@ulstu.ru xayc73@gmail.com mari-u@inbox.ru epira@mail.ru

Abstract. The article is devoted to the analysis of diagram processes based on finite state grammars. A semantic verification of diagrams on error using a timed automata grammar, and the method of checking diagram errors based on the extraction of ontological structures from the diagram are proposed.

Keywords: automata, grammars, visual language, workflow, parsing, patterns.

1 Introduction

Dynamic design process has become one of the latest trends in business process management (BPM). Both researchers and practitioners are improving the tools, techniques and theory of flexible design processes [1]. The basic definition of dynamic workflow in automata-based engineering as a flow of design work adapted to changes in the environment is given in [2]. Over the past 20 years, the management of the development of automated systems has solved the important tasks of creating, organizing, analyzing and monitoring workflows, as well as their implementation in the development of these systems. In [3], the authors give a detailed overview of the methods and tools of the organization of workflows, which includes the following stages: modeling of workflows "as is"; optimization by creating a model of the process "to be"; changing the organizational structure; "folding" workflows; process improvement. The application of systems in the development of workflows, such as ERP-systems, workflow management systems (WfMS), process modeling tools is discussed.

Flexible development of complex automated systems is associated with the adaptation of workflows to changes in system requirements. Enterprise flexibility is usually understood as the property of the enterprise to function in a dynamic world [4] and relates to two abilities: 1) to adapt to changes in the environment; 2) to open new opportunities constantly appearing in a dynamic world to launch completely new products (services). Becoming agile requires an approach that allows the system to identify and influence changes. The need to develop such an approach has arisen in response to the increasing pace of development needs.

2 Problem

In the theory of visual languages there are no effective methods of analysis and verification (especially in the part of full verification) of syntactic and semantic errors of diagram models. Methods the "direct" tests are implemented in most development tools, diagrams (e.g. MS Visio). These methods require multiple passes, which allows the system to analyze the syntactical correctness of the diagrams and identify the main errors. However, they cannot reveal context-dependent semantic errors associated with the text of any diagram model, denotative and significative semantics errors in terms of synonymy and antonymy, isomorphism of diagram models [22]. The above facts reveal additional types of "expensive" errors in the design of automated systems (as), which are difficult to diagnose. Analysis and verification of these errors is an important scientific and technical task.

3 Related work

Whitestein Technologies, Magenta Technologies, SkodaAuto, Volkswagen, Saarstahl AG note that the first generation of static (monolithic) systems for product life cycle management and project work flows [5] no longer meets the requirements of many companies. The approach and automated means of standardization of the first generation processes of project work flows have exhausted their resources, as a result there are unformalized (unsuitable, containing semantic errors) processes that stimulate the growth of the cost of their development and improvement.

The basic idea is characterized by completeness, relevance and consistency of the project is often lost due to the large amount of information in the development of complex automated systems using approaches based on Scrum, LSP etc. to ensure that the project met customer requirements and were successful, the designer proposes a conceptual model, giving an overview of the organizational structure of the project process with a description of the actors (entities) and their parts, functions and relationships between them.

There are conceptual models of the organization of flexible distributed flows of design work (SEO [2]), which have significant advantages over traditional models: SECI, schemes J.Zahman, Rational ADS, which are presented below. The traditional workflow management system is ProBis. According to [6] YAWL

Proceedings of the XX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2018), Moscow, Russia, October 9-12, 2018

(another document language), iPB is a flexible system for managing project work flows.

These systems use both diagram models of the working process and solve the problem of structural (syntactical) analysis and semantic (semantics) diagrams. For example, the authors [7] use colored Petri nets for dynamic semantic analysis of working processes, and in [8] use the approach of pi-calculus, formalizing working processes in algebraic expressions of the first order logic.

Development and research of workflow is based on visual languages such as UML, IDEF, BPMN, SDL, ER etc. In the modern theory of graphic visual language for the graph view uses the logical model. It contains both graphical objects and relationships between them. These models are processed using graphic grammars. The use of such a model for the verification and analysis of structural, i.e. topological (syntactical) features of diagrams is very difficult. This model is handled by graphical grammars. John L. Pfaltz and Azriel Rosenfeld proposed a two-dimensional model for generating web grammars [9]. Sentences generated by web grammars are oriented graphs with characters at the vertices ("cobwebs"). Zhang proposed positional graphic grammars [10, 13], later it was developed by Costagliola [11]. This is due to context-free grammars. Wittenberg and Weitzman [12] developed some relational graphic grammars. Zhang and Orgun [13] described the remaining graphic grammars in their studies. These methods have the following disadvantages.

1. Positional grammars, developed on the basis of plex-structure, do not use common points and cannot be used for languages whose objects have dynamically changing number of inputs/outputs.
2. There is no semantic check of text charts.
3. The common disadvantages of the above-mentioned grammars include: when designing grammar for unstructured visual languages, the number of rules increases (if the number of language primitives for describing all variants without structure does not change, the number of rules will increase significantly), the complexity of grammar development. In addition, the analysis of diagrams takes a long time (analyzers based on the considered grammars have polynomial or exponential analysis time).

In [14, 15] the authors propose a syntactically-oriented approach based on finite RV-grammars for visual language processing. However, there are no mechanisms of analysis and control of structural and semantic features of graphs from the point of view of their integrity and consistency between themselves and the conceptual model, including text attributes.

In the most common diagram model creation and processing tools, such as Microsoft Visio, Visual paradigm for UML, ARIS Toolset, IBM Rational Software Architect (RSA), analysis and management of diagram models is performed using direct methods that require multiple passes (this depends on the type of error monitoring). There is no control of structural features of integrated schematic models, as well as semantic control of integrity and coherence of structural and textual attributes of coupled schematic models of flexible working processes.

Thus, to solve the above problems, it is necessary to develop an integrated system of new models, grammars and tools for analysis and management of schematic models of flexible distributed workflows that provide a solution to the paper problem (task).

3 Definition for grammar. Semantic checking of diagrams on errors

The authors have developed automatic grammar, called RV-grammar, for the analysis and verification (control) of diagrams for the tools described in [16]. In collective design, diagram models have a complex hierarchical structure (complex diagrams), which increases the number of graphical elements many times. Classical graphic RV-grammar is complicated, the process of its development is complicated. There is no management of interconnected nodes and diagrams of complex models. For eliminating these disadvantages of the proposed multi-level grammar [23]. In a collective design for groups, it is important to control the consistency of the complex of the developed charts. Developers, accounting schema, you can forget to take into account a previous decision (as reflected in the schemes). For example, a Developer can drill down on a specific component of the system and lose focus on the fact that in previous stages such a component was implemented using other technologies. It is proposed to make a multilevel grammar for the analysis of its semantic correctness. The upper level of composition grammar will be the grammar of case diagrams, since the development of systems should start with this diagram (according to RUP methodology). When we start to use a definition like clock or timer, we can extend the RV grammar by adding the Clock parameter in this grammar. We can check 23 errors [24].

During the analysis, semantic information about the subject area will be preserved, and each new Diagram will be added to the overall structure on the basis of the successive expansion of the concepts of the subject area. It is consistent within the diagram, which is tested when developing the first semantic graphs. Then the use of semantic concepts in the pair is checked. When the developer adds the consistency of the diagram, the consistency of the complex model developed by the automated system is checked separately (see Figure 1). To check the complex model it is necessary to develop a graph of semantic links between the elements of the automated system. An adapted method of lexical and syntactic templates [25] was chosen to solve this problem. Currently, there are two main classes of methods for extracting knowledge from texts: statistical method and template-based method. Each of them has both advantages and disadvantages. The statistical method is mainly focused on data flow processing.

Methods of this class are considered the better, the greater the volume of the studied texts.

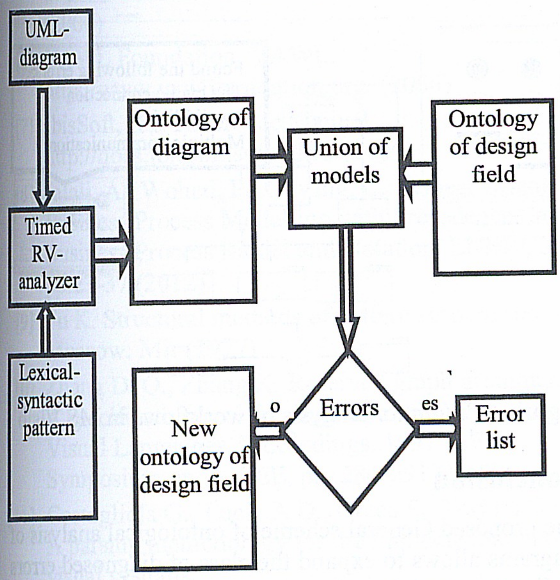

Figure 1 General diagram of the ontological analysis of any diagrams

Template-based methods use information about the target language to retrieve data. The methods of this class are characterized by the absence of dependence on the volume of the studied data volume. In connection with the importance of the diagnosis of errors at the early stages, we have selected the lexical-syntactic templates. Lexical and syntactical patterns allow the developer to create a semantic structure corresponding to the conceptual content of the text unit. For this reason, use is made of the features of the language of the text. In the analysis of the texts used the ratio of compatibility of text units. This ratio indicates the recording of some syntactic and grammatical properties of lexical units. In graphic grammars, the lexical unit is the term (sub-term) of grammar combined with text characteristics, immersed in it. The semantic graph of the diagram language consists of these lexical units. Let the element that represents the entity in the diagram be called a block, and the relationships between the blocks be called relationships. For clarity, the authors suggest that the blocks contain primitive concepts (single or composite). Let the blocks be NP and the relations be REL. there are 3 type of relationship (REL) chart of use: addition (inclusion) and the expansion of relations "part-whole"; aggregation-the relationship of "above-below"; communication - non-deterministic relations. Then the ontology taxonomy will be built on NP blocks. Ontological relations are based on the relationship REL. The following rule is used: NPi + rel + NPj. The use case diagram inclusion rule determines that NPj is a top-level NPj concept. In order to implement the modified method, it is necessary to modify the production rules of classical RV grammar. After you run the rule, you must run the procedure to extract semantic information from the graphical chart element. The rule is as follows:

$$\tilde{a}_t(\chi) \xrightarrow{W_\upsilon(\gamma_1,\ldots,\gamma_n)} r_m \qquad (1)$$

where χ is the procedure of semantic information extraction. The χ procedure is presented as a search for the corresponding rule in the list (for example, scenario 1). Rule (7) consists of two parts, a replacement and a pattern. Part of the pattern describes the lexical and syntactic pattern. Part of the block defines a quasi-term, part of the text defines the syntactical characteristics of the text in that quasi-term. The replacement part determines the location of the text unit in the partial semantic tree of the chart. For example, we proposed to check the diagram shown in Figure 2.

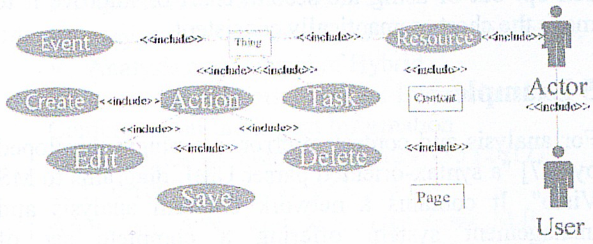

Figure 2 Use-case diagram

The example of rules for quasi-term Actor of this diagram can be written as Script 1.

Script 1 The rule (7) for quasi-term Actor

```
<rule>
      <pattern>
         <block>
            <type>actor</type>
         </block>
         <text>
            <type>noun</type>
         </text>
      </pattern>
      <replacement>
         <place>actor</place>
         <value>text</value>
      </replacement>
 </rule>
```

The following templates are used for this diagram:

block{type: actor} –> Resource > Actor [value: name]

block{type: usecase} & block{value: name} == verb –> Action[value: name]

block{type: usecase} & block{value: name} == noun –> Resource>Action[value:name]

Consider what is a block{type: usecase} & block{value: name} == verb –> Action[value: name] block. A condition consisting of two parts combined by the & character is written in the left part. This means that the pattern is applicable when both parts of the condition are satisfied at the same time. The first part of the condition means that the rule applies to the block of the type of use. The second part is the text in the block, which is a noun. As a result, the rule applies to a noun in the use case block. The second part shows the place in the hierarchy of the value of the rule. In this case, the rule means that the fragment selected at the first stage must be placed in the ontological network under the concept of action. Assume that the diagram that is shown below is described in the second step and the result is the following semantic network. When combining ontologies, the following main elements are highlighted: Action, Content, Actor. The result of the combination is shown in Figure 3. The analysis of this diagram shows that the network will fall into two parts. This result means that the diagrams describe different use cases and do not agree with each other. The reason for this result may be the following: the inclusion of the diagram from another project; a poorly designed scheme or absence of binding of the chart. In each case, the Developer receives an error message, and the second scheme is marked as incorrect. The developer can opt out of using the second chart or improve it to make the charts semantically consistent.

5 Example

For analysis and control of error is a plugin developed by [17] "a syntax-oriented parser UML diagrams to MS Visio". It contains a network diagram analysis and management system offering a complete set of functions for analyzing and controlling syntactic and semantic errors of diagrams (Figure 4). The plugin allows you to perform: analysis of any diagrams (BPMN, UML, IDEF0, IDEF3, Petri net) not only BPMN using Timed RV-grammar [24]; integrate as plugins in MS Visio; checks 23 types of errors [24]; gives recommendations to the designer to improve the chart.

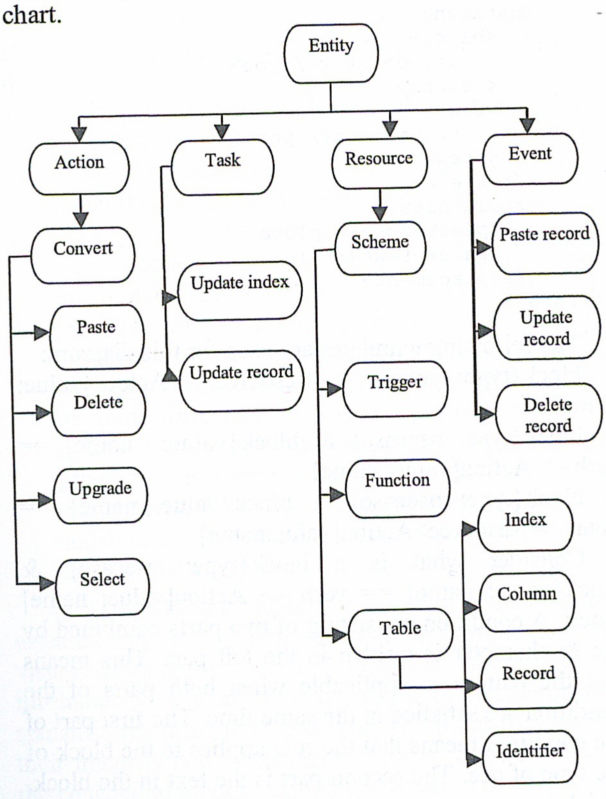

Figure 3 Use-case diagram

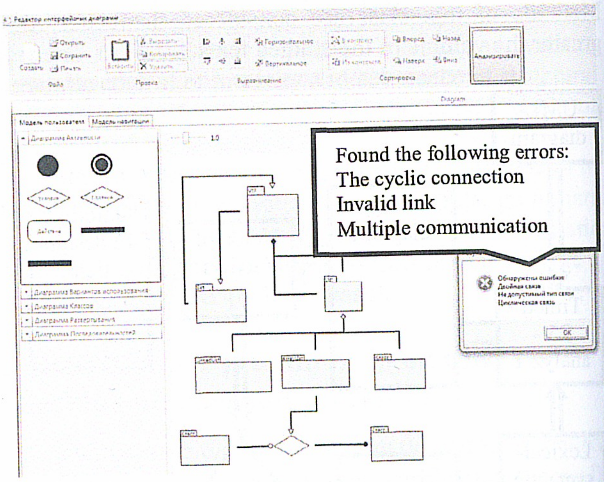

Figure 4 Plugin for analysis of workflows in MS Visio

Conclusion

The proposed General scheme of ontological analysis of diagrams allows to expand the class of diagnosed errors of diagram models and to organize a semantic check of the text of diagrams. The authors used Timed RV and Timed RVM-grammar for visual languages, which are actively used in as design practice: IDEF, eEPC, BPMN, SharePoint. In further works, it is planned to conduct experiments in eEPC and other graphic temporary languages for the development of a comprehensive system taking into account design data and technological preparation of the production of the real enterprise. We will identify new typical structural and behavioral errors and describe them in our future works.

Acknowledgments. The reported study was funded by RFBR according to the research project № 17-07-01417. 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] Thiemich, C., Puhlmann, F.: An Agile BPM Project Methodology. In BPM Conference. van der (2013)

[2] Shore, J., Warden, S.: The art of agile. O'Reilly (2008)

[3] Agile Business Process Development: Why, How and When - Applying Nonaka's theory of knowledge transformation to business process development. https://www.researchgate.net/publication/266078141

[4] Becker, J., Kugeler, M., Rosemann, M., eds.: Process Management: A Guide for the Design of Business Processes. 2nd ed. Springer (2011).

[5] Highsmith, J., Orr, K., Cockburn, A.: E-Business Application Delivery, pp. 4-17. http://www.cutter.com/freestuff/ead0002.pdf (2000)

[6] YAWL Foundation. YAWL. http://www.yawlfoundation.org/ (2004)

[7] IbisSoft, iPB Reference Manual. http://docs.ibissoft.se/node/3 (2009)

[8] Jalali, A., Wohed, P., Ouyang, C.: Aspect Oriented Business Process Modelling with Precedence. In Business Process Model and Notation, LNBIP, 25, pp.23-37 (2012)

[9] Fu K. Structural methods of pattern recognition, Moscow: Mir (1977)

[10] Zhang D. Q., Zhang K. Reserved graph grammar: A specification tool for diagrammatic VPLs. In: Visual Languages. Proceedings. 1997 IEEE Symposium on. – IEEE, pp. 284-291 (1997)

[11] Costagliola G., Lucia A.D., Orece S., Tortora G.: A parsing methodology for the implementation of visual systems. http://www.dmi.unisa.it/people/costagliola/www/home/papers/method.ps.gz.

[12] Wittenburg K., Weitzman L.: Relational grammars: Theory and practice in a visual language interface for process modeling. http://citeseer.ist.psu.edu/wittenburg96relational.html (1996)

[13] Zhang K. B., Zhang K., Orgun M. A. Using Graph Grammar to Implement Global Layout for A Visual Programming Language Generation System. (2002)

[14] Sharov O.G., Afanas'ev A.N.: Syntax-directed implementation of visual languages based on automaton graphical grammars. Programming and Computer Software. 31(6), 332-339 (2005)

[15] Sharov, O.G., Afanasev, A.N.: Methods and tools for translation of graphical diagrams. Programming and Computer Software. 37(3), 171-179 (2011)

[16] Afanasyev, A., Voit, N.: Multi-agent system to analyse manufacturing process models. In: Proceedings of International conference on Fuzzy Logic and Intelligent Technologies in Nu-clear Science, pp. 444-449 (2016). doi: 10.1142/9789813146976_0072

[17] Svidetel'stvo № 2016616685 Rossiiskaya Federaciya. RV-analizator diagrammnogo yazyka BPMN dlya MS Visio: svidetel'stvo o gosudarstvennoi registracii programmy dlya EVM / Afanasev A.N., Voit N.N., Kirillov S.Yu.; zayavitel' i pravoobladatel' Ul'yan. gos. tehn. un-t. № 2016616685; zayavl. 19.04.2016; zaregistr. 16.06.2016 (2016)

[18] 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

[19] Afanasyev, A.N., Voit, N.N., Gainullin, R.F.: Diagrammatic models processing in designing the complex automated systems. In: 10th IEEE International Conference on Application of Information and Communication Technologies (AICT), pp. 441-445 (2016). doi: 10.1109/ICAICT.2016.7991737

[20] Afanasyev, A., Voit, N., Timofeeva, O., Epifanov, V.: Analysis and Control of Hybrid Diagrammatical Workflows. In: International Conference on Intelligent Information Technologies for Industry, pp. 124-133 (2017). doi: 10.1007/978-3-319-68321-8_13

[21] Afanasyev, A.N., Voit, N.N., Ukhanova, M.E., Ionova, I.S.: Development and analysis of design-engineering workflows (mentioned as an instance a radio engineering enterprise). In: 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT), pp. 169-172 (2017).

[22] Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In: IJCAI-2001 Workshop on ontologies and information sharing, pp. 53-62 (2001). http://www.academia.edu/download/30556942/ontol2-proceedings.pdf#page=55

[23] Gainullin, R.: Ontology method of semantic analisys and control UML-diagramm. In: Interactive Systems: Problems of Human-Computer Interaction, pp. 122-128 (2013). https://www.researchgate.net/profile/Petr_Sosnin/publication/258517145_Interactive_Systems_Propblems_of_Human-Computer_Interactions/links/0c96052885745b45c4000000/Interactive-Systems-Propblems-of-Human-Computer-Interactions.pdf#page=124

[24] Afanasyev, A., Voit, N.: Grammar-Algebraic Approach to Analyze Workflows. In: International Conference on Computational Science and Its Applications, pp. 499-510 (2018). https://link.springer.com/chapter/10.1007/978-3-319-95171-3_39

[25] van der Aalst, Wil MP, et al.: Pattern-based analysis of BPML (and WSCI). In: QUT Technical report, FIT-TR-2002-05, 2002. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.7424&rep=rep1&type=pdf