Adaptation of the DFD-technology in the Modeling of Business Processes in the Environment of RDS

Georgiy Kalyanov[0000-0003-2429-0703], Boris Kupriyanov[0000-0002-0101-9104] and Olga Lukinova[0000-0002-5576-7749]

Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia
kalyanov@mail.ru

Abstract. The article is devoted to the description of structural methods of processes modeling, represented by visual languages of business processes modeling of socio-economic systems, designed not only for static functional/information modeling of processes, but also allowing to simulate their behavior. The description of grammar of structural languages of designing business processes of the enterprise which subsets it is offered to adapt for expansion of input language of the tool complex of RDS (Calculation of Dynamic Systems) that will allow to use opportunities of a complex for the solution of problems of the analysis not only dynamic, but also business models of organizational and economic systems are given. Provides the syntax and semantics of DFD, STD, ERD-notations. For the formal description of the syntax it is proposed to use the apparatus of mixed grammars, which are a combination of graph and ordinary grammars. The article describes the grammar that generates the simplest dialect of DFD-technology, informally describes the semantic aspects of the language, in particular the semantics of relations between the objects of the language. The types and sorts of evaluation criteria in the problems of business models quality analysis, syntax errors detection, as well as static semantics errors during their implementation in RDS are described.

Keywords: Modeling language, Business process, Tool and software complex of the RDS, DFD-technology, Combined graph grammars.

1 Introduction

Software complex RDS (calculation of Dynamic Systems) [1] is a tool for building research stands, providing the process of modeling, analysis and synthesis of control systems. The complex was implemented in the framework of projects which were implemented in the IPU RAS.

The article is devoted to the description of the language of modeling of business processes, based on the expansion of DFD-technology [2] and designed not only for static functional/information modeling of processes, but also allows to simulate their behavior. The possibility of such a simulation is due to the inclusion in the language of special structures, the translation of which into the input language of the RDS tool complex provides the necessary functionality. In addition, the implementation of the

proposed language allows to expand the capabilities of RDS due to its focus on modeling and analysis of BP.

The article presents the basic constructions of the proposed language, its syntax and semantics, the classification of potential errors. Implementation of the described methods and mechanisms is expected in the form of a supplementary module of simulation of business processes (BPSM) as part of a program complex of the RDS.

Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are identified in italic type, within parentheses, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

2 Purpose of the Business Process Simulation Modeling Module

We introduce the following concepts that require the purpose of this work.

Object O is the socio-economic system, carrying out various activities, activated by the need to obtain a certain result in favor of a certain subject C. the Initiation of an activity is carried out after the Input data set $A_i(a_1, a_2, \ldots a_n)$, associated with subject C and determining its state.

The relationship between subject C and object O is that the activity of object O, changes the state of subject C by changing the values of dataset $A_i(a_1, a_2, \ldots a_n)$.

Subject C can be both external to the object O, and internal, both physical and legal person, as some object, or process. Thus, the activity of the Object about acquires the character of subjectivity, which will build criteria for assessing the projected activity, the effectiveness of which is aimed at the interests of the subject.

Organizations, enterprises and other economic structures engaged in management, production, auxiliary and other types of activities are considered as the object of O. The activity is structured as follows [3, 4]:

1. The operation is atomic (indivisible) action that execute one job.
2. The function is a set of operations, grouped by a specific feature.
3. The business process is a related set of functions, in the course of which certain resources are consumed and a product is created (tangible or intangible result of human labor: object, service, scientific discovery, idea) that is of value to the consumer.
4. The subprocess is a business process that is a structural element of a business process and is of value to the internal client.

Then the business model is a structured graphical description of the network of processes and/or functions/operations related to data, documents, organizational units and other objects and entities that reflect the existing or proposed activity of the object O. Object modeling technology is based on structural visual languages:

- language of data flow diagrams (data flow diagrams, DFD-diagrams),
- object behavior modeling language (State Transition Diagrams, STD-diagrams),
- data representation language (ER-diagrams),

- language representation of the model system in the RDS environment.

Business process simulation module (BPSM) is a tool designed for:

- visual display of the activity model of the object O,
- model testing in the RDS environment to identify errors in static semantics.

BPSM is the software complex integrating module for:

- building and editing the business processes model of the object O,
- conversing the business processes model in a graphical language RDS,
- calculating of the dynamic system RDS that allows the analysis of the business model and simulation of its functioning.

Thus, within the framework of this stage, which is devoted to the issues of static processing of the business model in the RDS environment, the following tasks were solved:

1. Description of syntax and semantics of the necessary subset of structural languages in terms of transformation and representation of data flows, as well as behavioral aspects.
2. Representation business model (DFD, STD, ER-diagrams) in environment of RDS model.
3. Development of methods and algorithms of analysis and detecting errors of business model static semantics and syntax.

The work prospect is in the development of methods and tools for business processes formal description, which will allow to carry out its simulation modeling for dynamic errors identifying.

3 Structural Method of the Business Process Representation Based on the Graph Grammars and DFD-technology

Modeling languages used in the system BPSM include three types of basic building blocks of the business model: objects, relations and diagrams. In this case, objects are the basic elements (the alphabet of the language), the relationship links objects into semantic blocks (words), diagrams group words into "meaningful" phrases and sentences (business processes).

There are 3 types of objects in the language: functional object (process, subsystem, mini-specification, module, discriminator, etc.), information object (external entity, storage, information channel, entity, event, data area, etc.) and behavioral object (control process, state, etc.).

Relations between objects determine their interaction through information flows and control signal, or provide structural organization of object conglomerates (hierarchy, generalization, etc.).

A diagram is a collection of words represented as a directed graph with vertices corresponding to objects and edges corresponding to relations.

For the description of the above blocks as elements of the language of the BPSM system the apparatus of grammars is used. The classical grammar of the Chomsky language [5] is a tuple of four sets G = (T, N, P, s), where T is the alphabet of the language (the set of terminal characters), N is the set of grammar variables or nonterminal characters, P is the set of rules for deriving chains of the language L::=R

(L is a nonterminal character, R is a sequence of terminal and nonterminal characters), $s \in N$ is the initial character.

The formalism of graph grammars [6, 7] which are a generalization of Chomsky's grammars into graphs is usually used to set the syntax of visual languages. The role of traditional symbols in such grammars is played by graphic symbols or graphs/subgraphs.

In [3, 8] the model of DFD-technology (including a set of the above diagram techniques) in the form of a mixed graph with different types of vertices and edges for an adequate description of organizational and management systems was proposed, and a special parallel attribute generating grammar for the business process was developed, allowing to generate variants (scenarios) of its execution under various restrictions.

For the purposes of this work, an intermediate variant is proposed, it is a mixed grammar [2], the symbols of which can be not only graphs/subgraphs, but also fragments of visual models in various notations (within the framework of DFD-technology) up to the atomic symbols of the language. This corresponds to the introduction to the grammar of two types of terminal objects - detailed and non-detailed.

Thus, the specificity of graph grammars of structural languages is used to create business models of the object O is that:

- all terminal set T are graphical symbols,
- elements of the set T are represented by two types of objects: detailed (pseudo–terminal - terminal within a specific diagram) and non-detailed (terminal symbols),
- terminals are not only graphs /subgraphs, but also fragments of visual models in various notations, as well as other high-level descriptions.

In addition, such a grammar is a high – level structure, the upper level of which is a graph description, and the lower level is the classic output of chains of alphabetic characters. Set T is represented by the following symbols: PROCESS, MPROCESS, STORAGE, EXTERNAL ENTITY, FLOW, STATE, INITIAL STATE, FINAL STATE, TRANSITION, ENTITY, RELATION, DISCRIMINATOR.

4 Description of the Nonterminal Characters

In order to build a business model of the O object, this paper considers the following three types of nonterminal symbols:

- diagrams are represented of directed graph whose vertices are terminal objects $t_i \in T$ of the type “process”, “state”, etc., edges are terminal objects $t_j \in T, j \neq i$, of the type “relation”,
- minispecification, which is a detailed description of the logic of the final process,
- data dictionary, which determines the structure and content of all data flows in the diagrams.

Each of the nonterminals has a complex horizontal and vertical structure based on different types of relationships. They can be divided into two types: connecting objects at the same level of the model (relation – information flow, relation – control flow,

relation – transition, relation of communication); control inter-level relations (relations of decomposition of different types, the relation of categorization).

The semantics of the relations of the first type consists in data transmission or control (control signals) between entities of a specific level. In this case, the composition and structure of the transmitted data is determined by the appropriate grammatical rule. And the semantics of the control flow is determined by its type. The semantics of decomposition relations consists in inter-level balancing, i.e. in fact, in linking the first kind of relationship between the levels of the model, the basic rule of which is that all the relationships of the first kind associated with the detail object should be displayed (and linked to the corresponding objects) at the detail level. The categorization relation is the generalization relation.

The following is an informal description of these types of characters in the set N.

4.1 Description of Noterminal type "Diagram"

The nonterminal symbol "diagram" is represented by the following three groups:

- diagrams for illustrating the functions that the system must perform and the relationships between these functions, for this purpose, the DFD are used, supplemented by data dictionaries and lower-level process specifications;
- diagrams for data modeling and its relationships, ERD are used for this purpose;
- diagrams for system behavior modeling, for this purpose STD are used.

Diagrams of all three groups contain graphical and textual modeling tools. The first used to display the main components of the model, the second used to provide an accurate definition of its components and relationships.

4.2 Description of Nonterminal Type "DFD- diagram"

Nonterminals of this type are divided into two kinds: a context diagram, a diagram of data flows or control flows, the second is characterized by the presence of a special control process for behavioral control. In fact, it is analogous to the command post, sending commands to DFD-process with the help of control flows.

Nonterminal "Context diagram". The diagram has a star-shaped topology, in the center of which is the main process connected to the receivers and sources of information through which users and other external systems interact with the system. In this case, the context diagram may contain not a single main process, but a set of processes connected by data flows. The expediency of such detail is determined by the following factors: the presence of a large number of external entities); distributed nature of the object O; multifunctionality of the object with grouping of functions into separate subsystems. The context diagram is detailed by means of a decomposition relation.

Nonterminal "DFD- diagram". For each process existed in context diagrams, it is detail in the DFD-diagram or minispecification. The DFD-diagram shows are external to the system, the recipients and the sender of the data, identifies logical functions (processes) and the group of data elements connecting one function with another (flows), and identifies drives (storage) of data accessed.

After building a hierarchy of diagrams, the resulting model should be checked for the completeness of the original data system and the isolation of objects.

For executing DFD model in the RDS environment, it should contain pseudoterminal object MPROCESS, i.e. DFD-model needs to be complemented by the behavioral model.

4.3 Description of Nonterminal Type "STD-diagram"

The purpose of the nonterminal "STD-diagram" is to be able to model aspects of the business model of the O object that depend on the reaction to the event, i.e. STD is a tool for creating models of the behavior of the subject C. The nonterminal "STD-diagram" is in a hierarchical relationship with the symbol of the MPROCESS process and is a decomposition of the control processes of the DFD-model, and also describes the relationship between the input and output control flows for the control process. Thus, STD is used for controling of execution of functions of the DFD model.

The subject being modeled at any time is only in one of the finite set of states. If some condition is met, it can change its state, and transitions between states must be defined uniquely.

4.4 Description of Nonterminal Type "ER-diagram"

Nonterminal "ER-diagram" refers to the information objects of the business model. It defines how the internal structure of the STORAGE object is organized and contains information about the information entities of the business model and how they interact, identifying objects, the properties of these objects and their relationships with other objects in the model.

4.5 Description of Noterminal Type "Minispecification"

Each logical function (process) can be detailed using the lower-level DFD; when further refinement ceases to be useful, go to the expression of logic functions by using the process specification of the lower level (minispecification).

Minispecification is the object describing the logics of the process. It contains a process number, lists of input and output data, algorithm transformed input data flows in output ones. Minispecification is the lower of the hierarchy of the DFD model. The decision on the completion of detail of the process and the use of minispecifically is taken by the analyst on the basis of the following criteria: the process of a small number of input and output data flows; the process can be described in a sequential algorithm; the process performs a single logical function of converting input information into output; the logic of the process can be described as a small volume minispecification (no more than 20-30 lines).

4.6 Description of Nonterminals "Data Dictionary", Service Structures

A data dictionary is a service object organized in a certain way to store all data items with their exact definitions. The data dictionary defines the structure and content of all

data flows that are present in the diagrams. For each flow the dictionary stores: name, type, attributes. The data dictionary template is generated automatically when you create DFD objects.

Similar to the data dictionary, service structures (descriptors) are formed to store attributes, comments and so on for other chart objects.

5 Static Methods of the Business Model Analysis

The analysis of the business model of the object O is carried out in three directions: syntax, semantics and pragmatics. Analysis methods can be divided into the following classes:

- syntactical, i.e. those that reveal violations of the syntaxof the diagram,
- semantic - identify violations of the semantic representation of the object O and subject C, which is modeled by ER, DFD, STD-diagrams,
- quality analysis of the business model, evaluated by the parameters of coupling and cohesion.

5.1 Description of the Quality Analysis Criteria

Assessment of the business process quality [9] is based on the fact that the components of its business functions should be as independent as possible (the criterion of cohesion), and that each of them performs a single subtask (criterion of coupling). In a "good" business process, cohesion (as a measure of the interdependence of business functions) should be minimized, that is, functions should be as weakly dependent (independent) as possible. The method ranks the types of coupling business functions in order from weaker to stronger. In fact, the concept of cohesion generalizes the mechanisms of parameters transfer between the components of software systems and is only one of the criteria for assessing the quality of the business process split into parts: it assesses how well its business functions are separated from other.

Another criterion for assessing the quality of the business process dismemberment is the coupling criterion, which controls how the actions grouped in one function are related to other. Coupling is a measure of the strength of connecting functional and information objects within a single business function. Placing strongly related objects in the same function reduces cross-functional relationships and interactions.

The business process grammars [3] allow to build options for business process implementing, use a specified type of coupling. To solve this problem, the mechanism of synthesized attributes and special methods of their synthesis is used [3].

The proposed method of quality assessment allows to solve the following problems:

- define the types of business process coupling and cohesion,
- analysis of the identified types of coupling and cohesion,
- generation of variants of a business process with the specific types of cohesion and coherence.

5.2 Classification Scheme of Errors

According to the classification scheme, all errors (both syntactic and semantic) are divided into three classes: general errors, errors within each type diagrams of (DFD, ERD, STD), inter- diagram errors (interfaces).

Common errors can potentially occur in the construction of any diagrams of the language and relate to the syntactic aspects of its structures, for example, names.

Errors in DFD are divided into the following groups:

- objects do not link the information/control flows in accordance with the syntax of the language: storage – storage; storage - external entity, storage – control process, external entity – external entity, external entity – control process;
- isolated objects (no incoming /outgoing data flows): storage, external entity, process, control process;
- error in group flows: flow structure does not match the contents of the dictionary data, incorrect splitting/merging of flows;
- errors in the readings/records storage;
- process looping.

Errors in STD are divided into the following groups: the presence of isolated states, the presence of unreachable states.

Errors in ERD are divided into the following groups: absence of attributes in the entity, not a unique entity/attribute name, invalid types of relationships (many-to-many, recursive), the presence of isolated entities, the presence of non-binary relationships, incorrect division into subtypes (crossing subtypes, etc.).

Inter- diagram errors are divided into the following groups:

- DFD-DFD: vertical balancing (not matching cross-border flows by number, name, and structure);
- DFD - minispecification: Minispecification must present the names of all the incoming and outgoing flows;
- DFD - STD: STD action should correspond to DFD processes;
- DFD-ERD: control the correspondence of entities/attributes in ERD to streams/sub-streams in DFD (by names and types), control the number of stream/sub-stream attributes in DFD and ERD;
- DFD – STD: there are DFD control flows are owned by different STD;
- Context diagram –DFD: vertical balancing.

6 Conclusion

The article is devoted to the description of structural languages for business process design, which adapted to expand the input language of the RDS tool environment, which will allow to use the capabilities of RDS to solve the problems of analysis of business models of organizational and economic systems. Classification of evaluation criteria in the tasks of business model quality analysis, syntax errors detection, as well as static semantics errors in their implementation in RDS are described.

References

1. M.H. Dorry, A.A. Roshchin Program complex for modeling and research of control systems “Calculation of Dynamic Systems” (RDS). Part 1: RDS Device and circuit editing / / M.: LENARD, 2018.
2. G.N. Kalyanov, A conceptual model DFD-technology // Open Education, 2017, no. 4, pp. 21-26.
3. G.N. Kalyanov, Theory and practice of business process reorganization // M.: SINTEG, 2000.
4. G. N. Kalyanov, On the theory of business processes / / Software Engineering, 2018, vol. 9, № 3, p. 99-109.
5. A. Aho, J. D. Ullman, The Theory of parsing, translation and compiling. M.: World, 1978.
6. J. Rekers, A. A. Schuerr, Graph grammer aproach to graphic parsing // Visual Languages Proc., 11th IEEE Int. Symp., 1995, p.195-202.
7. D.Q. Zhang, K. Zhang, J. Cao, A context-sensitive graph grammar formalism for the specification of visual languages // The Computer Journal, 2001, vol. 44, № 3, p.186-200.
8. G.N. Kalyanov, Formal methods to support the reengineering of business processes // Economy, Statistics and Computer Science, 2013, № 3, pp. 161-165.
9. G.N. Kalyanov, Verification of business processes / / Proceedings of the 21st scientific and practical conference “Enterprise Engineering and Knowledge Management”. M.: 2018, vol. 1, p. 72-75.