# ANALYSIS OF UML NOTATION OF MODELING BUSINESS PROCESSES

[a]ANTON NIKOLAEVICH KARAMYSHEV[b]VERONIKA MIKHAILOVNA KRUTIKOVA[c]DMITRY FEDOROVICH FEDOROV[d]ILNUR ILJAZOVICH MAKHMUTOVe ILNAR AZBAROVICH MURTAZIN

*Kazan Federal University, 18 Kremlyovskaya street, Kazan 420008, Russia*
*Email: [a]antonkar2005@yandex.ru. [b]info@ores.su.*
*[c]russia@prescopus.com. [d]editor@ores.su.*

Abstract. Optimization of the main and auxiliary business processes is one of the most important factors in the competitiveness of large industrial enterprises. Optimization measures are developed based on the analysis of business process models to identify unnecessary and duplicate operations, eliminate unjustified losses of all kinds, reduce working capital and shorten the production cycle. This article analyzes one of the most common business process modeling technologies - uml notation. The types of models that are developed within the framework of this technology (diagrams of use cases, packages, activities, states, cooperations, sequences, deployment classes, ), four types of entities used in diagrams are considered: 1) structural (class, interface, component, use case, cooperation, node); 2) behavioral (interaction, automata); 3) grouping (packages); 4) annotation (comments); distinctive features of uml notation and application specifics, stages of its development. To demonstrate the most popular type of uml notation models, a private business process model has been developed.

Keywords: uml notation, modeling, business process, optimization, process management.

## 1 Introduction

One of the most common technologies for modeling business processes today is uml technology. The widespread use of the uml language has contributed to its versatility and usability. Within the framework of uml technology, it is possible to develop several tens of types of models, however in practice no more than ten are regularly used (in uml terminology models are called diagrams). These include diagrams: use cases, classes, states, activities, sequences, cooperation, components, deployment.

## 2 Methods

Table 1 shows the main purposes of these diagrams (Leonenkov, 2016; Kulyabov & Korolkova, 2005; Shmuller, 2005).
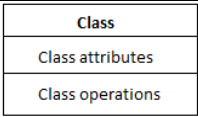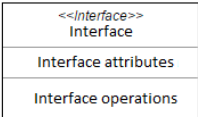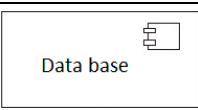
Table 1: Purpose of the most requested uml-notations

| Notation description | Notation main purpose |
|---|---|
| Use-case diagram | Describes business processes, reveals their interconnection and interaction |
| Package diagram | Describes packages interrelations |
| Activity diagram | Describe dynamics of the life cycle of business processes taking into account their relationships, resources used and output results |
| Statechart chart | |
| Collaboration diagram | |
| Sequence diagram | |
| Class diagram | Reflects a package of interrelated objects, the internal state, architecture of objects and the relationships between them, data and objects structures |
| Deployment diagram | Reflects technological resources of an organization |

*the genesis of uml technology.* In the 80s, object-oriented technologies began to be actively used in process management. To visualize business processes, special methods were developed (booch, oose, omt, and others), however, none of them allowed conducting full-fledged modeling of business processes. Achieving this goal was carried out by the synthesis of all these methods in the framework of one uml methodology. Over four decades, the methodology has been developing and on its basis, the modern unified modeling language uml (unified modeling language) was formed (Buch et al., 2006).

*the structure of the uml language.* The modeling process in uml is an indication of the existing relationships between entities (each entity uses its entity). Entities are the basic elements of a model. Four types of entities are distinguished: 1) structural (class, interface, component, use case, cooperation, node); 2) behavioral (interaction, automata); 3) grouping (packages); 4) annotation (comments). Each entity has its graphic representation, due to which an unambiguous understanding of the model is achieved. Consider the essence of the uml language (table 2) (Kulyabov & Korolkova, 2005; Matzyashek, 2016; Kimmel, 2008; Rosenberg, D. Scott, 2002).

Table 2: Entities in uml models

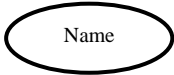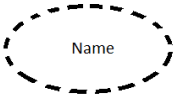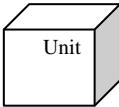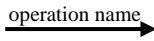| № | Entity Designation | Graphic Representation | Description |
|---|---|---|---|
| 1 | Structural | | Static parts of the model corresponding to the conceptual or physical elements of the system. Formulated by substantives in uml models. |
| 1.1 | Class | **Class** / Class attributes / Class operations | Collection of elements having common attributes, operations, semantics |
| 1.2 | Interface | <<Interface>> Interface / Interface attributes / Interface operations | Series of operations characterizing the behavior of individual elements of the model. |
| 1.3 | Component | Data base | Substituting system fragment that implements a set of interfaces. |

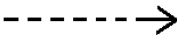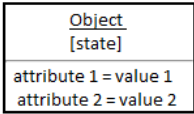| 1.4 | Use case | Name | Group of actions reflecting the interrelation of the system and the user |
|---|---|---|---|
| 1.5 | Cooperation: | Name | Series of interacting elements of the system that perform a specific task |
| 1.6 | Unit | Unit | physically existing system element that has a certain computing resource (for example, network equipment: switches, gateway) |
| 2. | Behavioral | | Dynamic parts of the model that describe the behavior of objects in diagrams. Formulated by verbs in models. |
| 2.1 | Interaction | operation name | Message exchange between the objects (ie entities 1.1 - 1.6) to achieve the specified goal. Includes messages, actions-connectors (provide connections between objects) |
| 2.2 | Automatic | State | The sequence of states  (situation in the life cycle of an object, during which it meets certain conditions, performs certain activities or expects certain events),  through which the object passes during its life.   It includes states, transitions, and events. |
| 2.3 | Activity | Action | State that describes the sequence of actions |
| 3. | grouping | | Block connecting objects with common characteristics together |
| 3.1 | Packages | Package name | The entity used to group related elements (for example, school class package includes such elements as a.a. Ivanov, ia petrov, etc.) |
| 4. | Annotative | | Explanatory components of uml models |
| 4.1 | Notes | Note | The explanatory note to separate elements of the model |
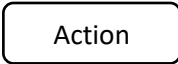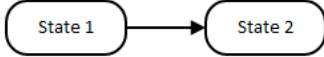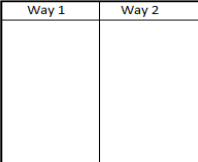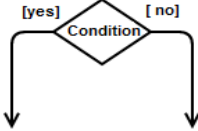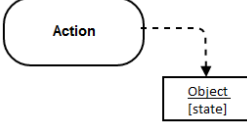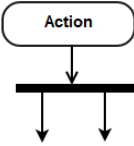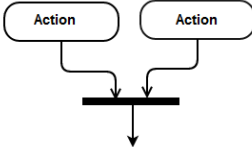
Table 3. Relationships in uml models

| № | Relation | Graphic Representation | Description |
|---|---|---|---|
| 1 | Dependence | | Type of relationship in which a change in one entity will entail a change in the characteristics of another entity |
| 2 | Association | | Type of relationship where objects have logical or semantic connections |
| 3 | Generalization | | Type of relationship in which the object of the private element (child) can be represented as a generalizing element (parent) |
| 4 | Implementation | | Type of relationship in which one entity defines the rules (instructions) for executing another entity This type of relationship is used: - between interfaces and classes; - between the use cases and the cooperatives that realize them (see p. 1.5 of table 2). |

In practice, the most commonly used uml model is an activity diagram. Consider her.

The activity diagram shows the set of actions necessary to achieve a specific goal. The advantages of this type of diagrams are the ability to depict parallel processes and detail complex algorithms. The main objects of the activity diagram include action, transitions, track, object, branching, object trajectory (table 4) (Kimmel, 2008; Craig, 2004; Jacobson & Ng, 2004; Koznov, 2007; Makhmutov et al., 2016; Galiev, 2017).

Table 4. The main objects of the activity diagram

| Activity diagram element | Graphic Representation | Element characteristics |
| --- | --- | --- |
| Action | Action | Named element that characterizes one of the stages of activity |
| Transitions | State 1 → State 2 | Reflect the relationship between two states and determine the conditions for the object transition from one state to another |
| Way | Way 1  Way 2 | Defines the boundaries of the work performed within the model |
| Object | Object [state] attribute 1 = value 1 attribute 2 = value 2 | Result of action |
| Branching | [yes] [ no] Condition | Due to this element, the process branching is carried out to consider possible scenarios |
| Object trajectory | Action ⇢ Object [state] | Attachment of an object to an activity using a dependency relationship. |
| Separation | Action | Used to separate parallel flows |
| Merger | Action   Action | Used to connect parallel flows |
| Initial state | ● | Indicates the beginning of the process |
| Final state | ◉ | Indicates the beginning of the process |

As an example, we simulate the "ticket purchase" business process in the activity diagram (fig. 1).
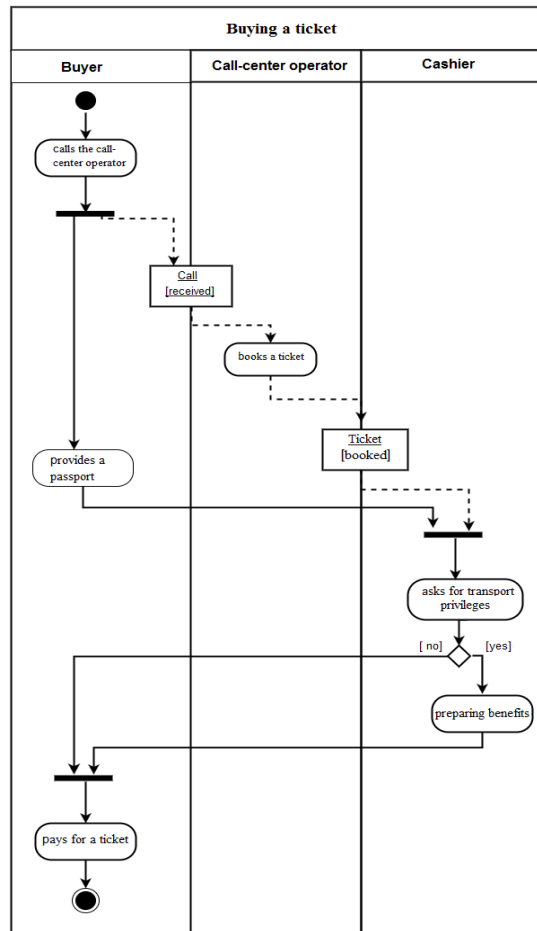
Figure 1. Diagram of activity of the business process "buying a ticket "

Three tracks of the activity diagram (buyer, call center operator, cashier) specify their area of responsibility. The buyer calls the operator, the "call" object is the result of this action. Finding this object at the junction of the tracks shows the interaction of two participants with each other. When booking a ticket and issuing passport data, the two flows of activity merge. The registration of transport benefits depends on the availability of the right to receive them from the buyer, i.e. At this stage, the process branches in the model. The final step is to pay for the ticket by the buyer. The hierarchical architecture of the model shows the sequence of actions to achieve the ultimate goal - buying a ticket.

Distinctive features of the uml language. The main distinguishing features of the uml language from others are direct and reverse code generation with the possibility of subsequent use in various programming languages.  Based on the uml graphical model, direct code generation is performed. Reverse generation is the process of forming a model based on program code. The uml standard supports code generation for all types of diagrams. The transformation of the graphic model into program code is carried out using special tools, which are called case – tools. There is a large selection of case tools that are used with uml. Each user can choose the most convenient tool for their work. Representatives of such software products include: "ibm rational rose", "umbrello uml modeller", "borland together", "microsoft visio", "sparx systems enterprise architect", "gentleware poseidon" (Matzyashek, 2016; Fowler, 2006; Konaplen, 2001; Ivanov & Novikov, 2010; Makhmutov et al., 2016; Galiev, 2017).

When choosing case-tool, the developer must correlate the goals of the object-oriented modeling project with the features of the case-tool.

You must pay attention to the following points:

A) the possibility of code generation in the most common programming languages.

Part of the case-tools (usually with limited capabilities) does not allow code generation in the most common programming languages (for example, in  "ada", "java", "c", "c ++", "basic" and other). These include, for example, rational rose modeler. More functional software products "rational rose professional", "rational rose real-time" provide such opportunities.

One of the editions of "gentleware poseidon enterprise edition" provides a wide range of features: documentation generation in html and rtf formats, support of such programming languages as "c ++", "java", "c #", "visualbasic", "visualbasic.net", " delphi "," php ",".net ", uml 2.0 notation with support for all kinds of diagrams (Shmuller, 2005; Koznov, 2007).

B) the ability to support the necessary types of uml diagrams and synchronize the encoding with the graphic model.

One of the few programs that implement this feature are the programs "borland together" and "gentleware poseidon enterprise edition" (Leonenkov, 2016; Buch et al., 2006; Konaplen, 2001; Vendrov, 1998).

*uml application*
Business process modeling is carried out in leading global corporations. The use of process technologies allows improving the management of individual business processes and the enterprise as a whole. The economic performance of companies is improved by building optimal value chains, reducing working capital, eliminating unnecessary and duplicate functions, and automating business process models in the enterprise's information system.

Specific features of uml notation, including the ability to integrate and convert uml encoding into the most common programming languages, have determined its widespread use. According to the results of research by the state of business process management-2016 conducted by bptrends (Harmon, 2016), uml notation with 17% is the third most widely used by leading global companies (bpmn notation first with 64%, second - aris notation with 18%).

**3 Results and discussion**

In the course of the study, the following methods were applied:

1.  Selective analysis of specialized literature with a high citation index on the topics indicated in the title of the article. In particular, information was collected on uml notation diagrams, objects, and relationships in these diagrams.
2.  The generated array of information was systematized for further analysis. For a better understanding, examples of business processes in the most common activity diagram were developed.
3.  The results of the study were given the author's interpretation, conclusions are drawn. Uml notation allows you to visualize enterprise business processes in several dozen types of models, but in practice, about ten types of models are used. The most common is the "activity diagram" model, which allows you to specify the areas of responsibility of all participants in the process and determine their object relationships, describe possible scenarios of the business process.

**4 Summary**

A significant advantage of uml technology is direct and reverse code generation with the possibility of subsequent integration into the enterprise information system. At the same time, modeling can be carried out in several different software environments; the choice of modeling tools is determined based on the ultimate tasks of implementing process control.

**5 Conclusions**

This study put effort to optimize the main and auxiliary business processes as one of the most important factors in the competitiveness of large industrial enterprises. In this regard the analysis of business process models to identify unnecessary and duplicate operations, eliminate unjustified losses of all kinds, reduce working capital and shorten the production cycle are developed to further investigate the goal. One of the most common business process modeling technologies which is known as uml notation using uml method allowed authors to visualize enterprise business process in numerous types. In this article the" activity diagram" model has been chosen among all existed models.

**Acknowledgments**

**Literature:**
1.  Leonenkov, A. V.: Notation and semantics of uml. Moscow. Niu "intuit". 2016. P.205.
2.  Kulyabov, D. S. Korolkova, A. V.: Introduction to formal methods for describing business processes. Moscow: rudn. 2008. P. 173.
3.  Shmuller, D. Learn uml 2 by yourself in 24 hours. Practical guidance. Moscow: williams. 2005. P.416.
4.  Buch, G. Rambo, D. Jacobson, I. The language of uml. User guide moscow: dmk press. 2006. P. 496.
5.  Matzyashek, L. Analysis and design of information systems using uml 2.0. Moscow: williams. 2016. P.816.
6.  Kimmel, P. Uml basics of visual analysis and design. Moscow.: nt-press. 2008. P. 264.
7.  Rosenberg, D. Scott, K. Application of object modeling using uml and analysis of precedents. Moscow: dmk press. 2002. P. 496.
8.  Craig, L. Application of uml and design patterns. Moscow.: williams publishing house. 2004. P. 624.
9.  Jacobson, I, Ng P. W. Aspect-oriented software development with use cases (addison-wesley object technology series). Addison-Wesley Professional; 2004 Dec 1.
10. Koznov, D. Fundamentals of visual modeling. Moscow. : internet-university of information technologies; binomial. Laboratory of knowledge. 2007. P. 248.
11. Fowler, M. Uml. Fundamentals of spb.: symbol-plus. 2006. P. 192.
12. Konaplen, D. Development of web-applications using uml moscow: williams. 2001. P. 288.
13. Ivanov, D. Yu. Novikov, F. A. Basics of modeling on uml spb. : publishing house polytechnic. University. 2010. P. 249.
14. Vendrov, A. M. Case-technology. Modern methods and means of designing information systems [Case–tehnologii. Sovremennye metody i sredstva proektirovanija informacionnyh sistem], Finansy i statistika, Moscow. 1998.
15. Harmon, P. A BPTrends Report «The State of Business Process Management. 2016.
16. Makhmutov, I. I. Isavnin, A. G. Karamyshev, A. N. Sych, S. A. Classification approach in the determination of knowledge in the context of organization. academy of strategic management journal. Vol. 15, is.special issue. 2016. pp. 40-46.
17. Galiev, D. R. Isavnin, A. G. Makhmutov, I. I. Portfolio investment models with asymmetric risk measures and using genetic algorithms. turkish online journal of design art and communication. Vol. 7, 2017, pp.1652-1662.

**Primary Paper Section:** A

**Secondary Paper Section:** AE, AH