Информационные технологии

UDC 681.518.2+658.511

The Development of Enterprise Modeling Instrument for Corporate Management Systems

G. M. Novikova, T. V. Malyutina

Information Technologies Department
Peoples' Friendship University of Russia
Miklukho-Maklaya str., 6, Moscow, Russia, 117198

This article deals with the research of enterprise modeling instrument's disadvantages and its ways of possible development. There is an example of an integrated notation, containing such notions as "Event" and "Condition", and also an example of using the formal level of application domain to estimate the completeness and the consistency of the models mentioned.

Key words and phrases: modeling, ARIS, enterprise modeling instruments, integrated notation, logical models, formal models.

1. Introduction

One of the most important challenges every company faces is designing organizational and functional structures, developing mechanisms, instruments and processes of management to raise the business effectiveness. Object modeling is to estimate design decisions. It helps to find a proper decision while designing the initiatives above and thus improving them.

At the present time there is a great number of methodologies, methods, models, languages and tools to describe activity of an enterprise due to different perspectives. For example, the Zachman framework divides the application domain according to the aspects of stakeholders. TOGAF (The Open Group Architecture Framework) assigns four major architectures: business architecture, applications architecture, data architecture, technology architecture.

However, nowadays enterprise modeling tools maintain only conceptual level and formal level is hardly represented. The expressiveness of all supported models doesn't have sufficient power to represent all aspects, laws, limitations and patterns of such an application domain as corporate management systems. There is no universal method of solving a problem that appears in the process of modeling and describing corporation and its systems in the development of visual tools, which aim to improve representation knowledge in graphical form. At the same time the method of modeling information systems and databases described in State Standard [1] and based on interpreted predicate logic, can't be applied to the description of an application domain and to the modeling of various process management problems.

2. The Disadvantages of Modern Instrumental Modeling Environment

Nowadays the most developed methodology and instrumental modeling environment is ARIS (Architecture of Integrated Information Systems) by IDS Scheer, the company, which represents the result of mergering of two companies: Software AG and IDS Scheer. ARIS is also the implementation environment, which allows to analyze and optimize business-processes by simulation. The following should be mentioned: the environment supports all existing methodologies and contains UML, used to classify the objects of the application domain.

Received 30th June, 2011.

Although nowadays integrated modeling environment ARIS is the most developed tool for business process modeling, the visual representation isn't complete. Particularly, there is no visual representation and semantic interpretation of such entities as "Condition" and "Situation". Modeling environment provides a wide range of entities, such as organization unit, employee, position, function and event, but there is no formal semantics. Among important questions of control modeling processes are their interconnection and synchronization, not fully described by existing tools of enterprise modeling.

Today ARIS maintains special tools that aim to estimate the correctness of application domain description. There are a lot of problems, such as duplication of organization units and functions while describing organization-function domain. But the solution of this problem remains local and is limited by the organization-function model. Expressive power of modeling environment and its functional capabilities isn't enough to verify exhaustiveness and consistency of application domain description in general.

Event-trigger chain of processes, supported by EPC (Event-driven Process Chain), doesn't take into consideration semantics of application domain, when event and action can differ in nature. For example, action can be interaction, which changes condition of an object, or a design transaction. Event also can obtain qualitative nature, which means changing condition of an application domain, or temporal nature as well.

Nowadays formal models, which describe functions and processes, are limited to input/output data.

Eventually such characteristics, as the type of a function, the objects it works with, what changes during the implementation occur, are missed.

3. Line of Development for Enterprise Modeling Instrument for Corporate Management Systems

Thereby there is a need to develop both methodology and modeling environment. That's why the following steps should be taken:

1. To extend the set of notion's types, used during the process of modeling company's activity, including the terms of enterprise and financial management, like situation, control action, cost driver and so on.
2. To implement enterprise anthology into the modeling environment to let it use set notions of application domain during modeling process, particularly notions from enterprise management field. First of all, classification should contain the following terms: notion, characteristic, event, condition, link, relation, operation, action, influence, process.
3. To widen the set of link types to connect application domain terms by sorting out structural, functional, logical and reflecting operation dynamics ties. The establishment of logical links among application domain terms allows to extend the number of factors, occurring during the process of logical inference, therefore the modeling environment has to support logical inference mechanisms, particularly resolution rule.
4. To separate the theory and the model of an application domain. The theory should contain closed logic formulas. Having theory's axioms turning true, the model stands for interpretation to those formulas. Closed logic formula is a law to follow. For instance, see the law of product profitability: the production price is lower than the use value. In that case the following problems that can be solved in the process of modeling usually exist:
   (a) mapping the theory and interpretation to see if this particular interpretation is a model
   (b) determining the theory for specified interpretation. For instance, given interpretation is the model of condition classes of pre-crisis situation
   (c) searching for interpretations, which are the models for the theory

5. To implement aids and tools for operating with linguistic variables: sell-through, good project, effective production, effective process, optimal manufacture, friendly interface.
6. To support the identification of software's definition level, specifications level and so on – the equilibration according to the level.

4. The Example of Integrated Notation

The integrated notation is a combination of ARIS's EPC diagram and UML sequence diagram.

Representing the combination of simplicity, clarity and functionality, the eEPC diagram is one of the most frequently used diagrams of ARIS [2]. However, there is a place for imperfection. The eEPC diagram contains two major entities: "Action" and "Event". A process chain begins and ends with an "Event". It is a body of the "Action" process ("Event" alternate) and there is no way for two entities of one kind are placed back-to-back. To apply common sense strategy such a system is correct, because

This system also helps to facilitate the analysis of a process, considering every "Event" as a peculiar checkpoint. But, in spite of all advantages, "Event" often becomes the complete version of "Action", only overloading a diagram.

Showing the example of using eEPC, Fig. 1. contains the description of the process of searching a person to fill a vacant position. There are three possibilities:

1. There is an appropriate employee in your company, who can fill the gap.
2. There is an employee in your company, but he needs to be trained.
3. There are no appropriate employees in your company. In this situation there are two ways: to apply to HR agency or start searching by yourself.

UML instruments can propose another solution by using a sequence diagram [3]. It shows objects interaction in time. UML as abundant instrument allows to show not only the interaction but also the condition of the objects taking part in the process. There exists the same process of searching a person to fill a vacant position on Fig. 2. Notion "Condition" was added to the diagram.

In spite of all the advantages of UML, its main disadvantage is poor visualization. A simple diagram is easy to read anyway, but a huge one will be unreadable. Calls and replies, which should be introduced in a diagram, overload the representation with arrows. Also the UML diagram, as opposed to eEPC, can't be expanded with details without overloading the diagram.

As a result, there is an opportunity to add the notion "Condition" to eEPC to upgrade the diagram. The condition of an object is a significant part of any process, because the changes in object's condition can be the initiator of the new process chain. There is transformation of "Condition" into "Event", which shows the changes in object's condition, in ARIS diagrams (Fig. 3).

The new element is divided into two parts: the upper part contains the name of the changing object; the lower one contains its condition.

Thereby, the integrated notation is going to be both visual sequence of procedures and condition changes sequence of objects in process.

The diagram shows, that the object, when changing its condition, can be in the same condition over a period of several actions. Also there can be seen the condition of position, marked "waiting" till the moment of formation of labour contract in the diagram (Fig. 4). This factor should be taken into consideration by the models, containing this object. For instance, the condition "waiting" requires distributing of position's functions and delegation of organization unit's authority during the process of finding and drawing up a labour contract.

Management of process execution consists of analysis of labour condition and decision-making concerning the way of process realization. The condition of an employee is described by his level of competency. The identification of labour level of competency should be examined as analysis point of business process. The decision

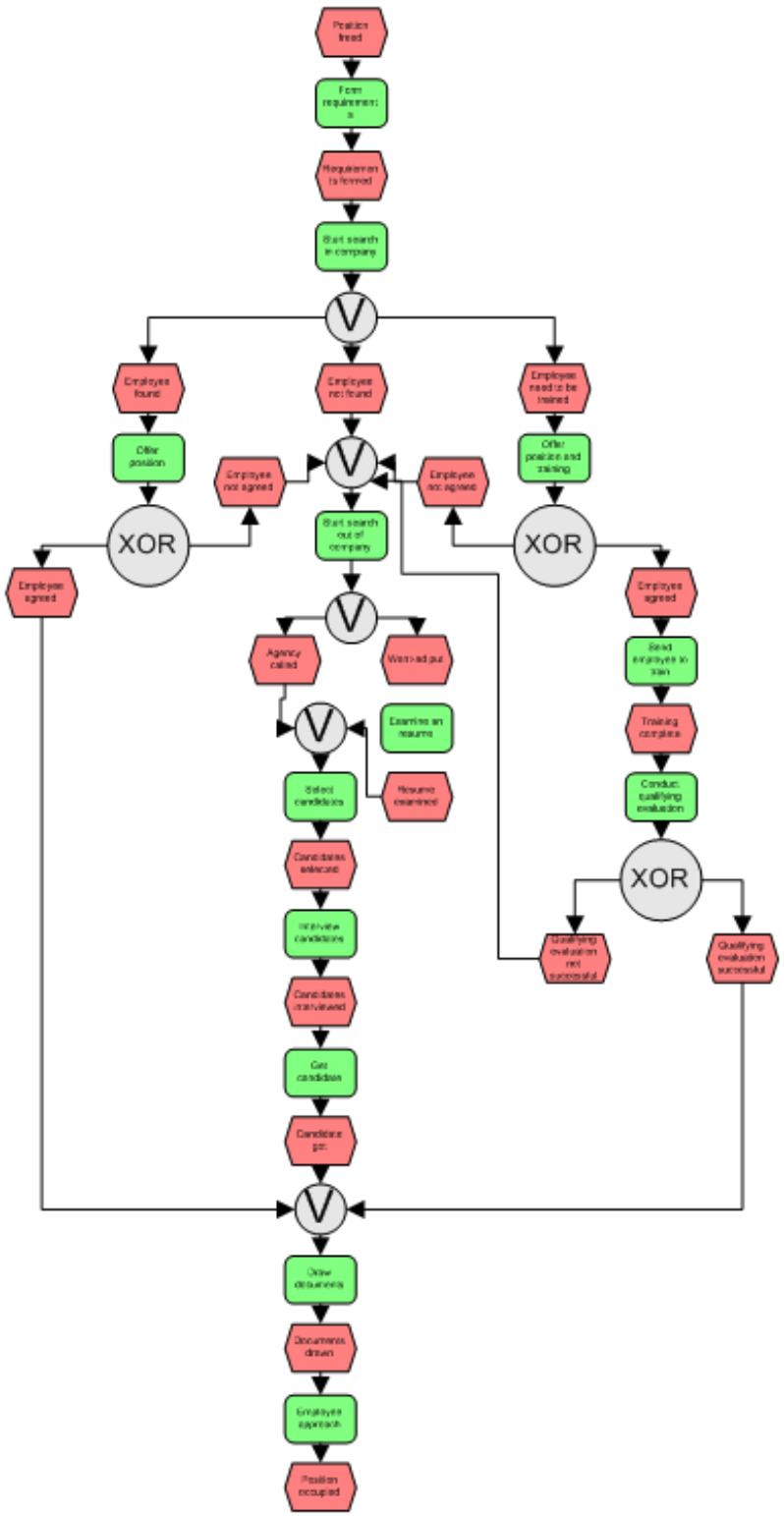

Figure 1. **ARIS Canonical Model**

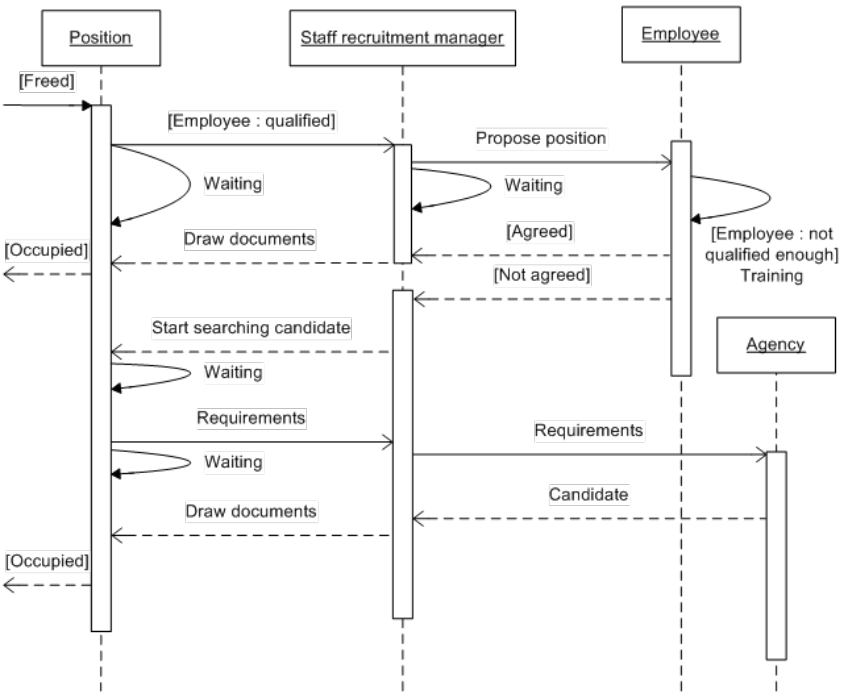

Figure 2. **Sequence Diagram**

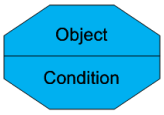

Figure 3. **New element "Condition"**

point is a moment of choice among the ways of searching candidates to fill the position outside the company. The control point of the business process is the comparison of a trained worker's skills with vacant position's requirements. (Fig. 5–7)

Thereby specification of the process should contain such parameters as analysis points, objects of analysis, parameters of analysis, control points, decision points.

Set input parameters of the process, such as current condition of an object, type of position and vacant position's requirements, the business can be focused on the current condition of the application domain.

5. Formal Level of Application Domain Description

There is not enough power not only to describe the application domain, but to construct the consistent and optimal model in the groups of ARIS's models, represented by Fig. 8. As a program is supposed to conform its specification, the same way models are supposed to conform the theory, which restricts the set of models, representing the interpretation due to true theory's axioms.

Apparently, one of the ways to detect the consistency and optimality of the model is simulation, but such a method requires considerable expenditure of resources let alone the fact that nonoptimality and discrepancy can be detected with a huge delay. At the same time there is an opportunity to mark out model's characteristic and define requirements for developing the model while designing it Requirements and rules of processing should be applied not only to the model's objects, but also to its links.

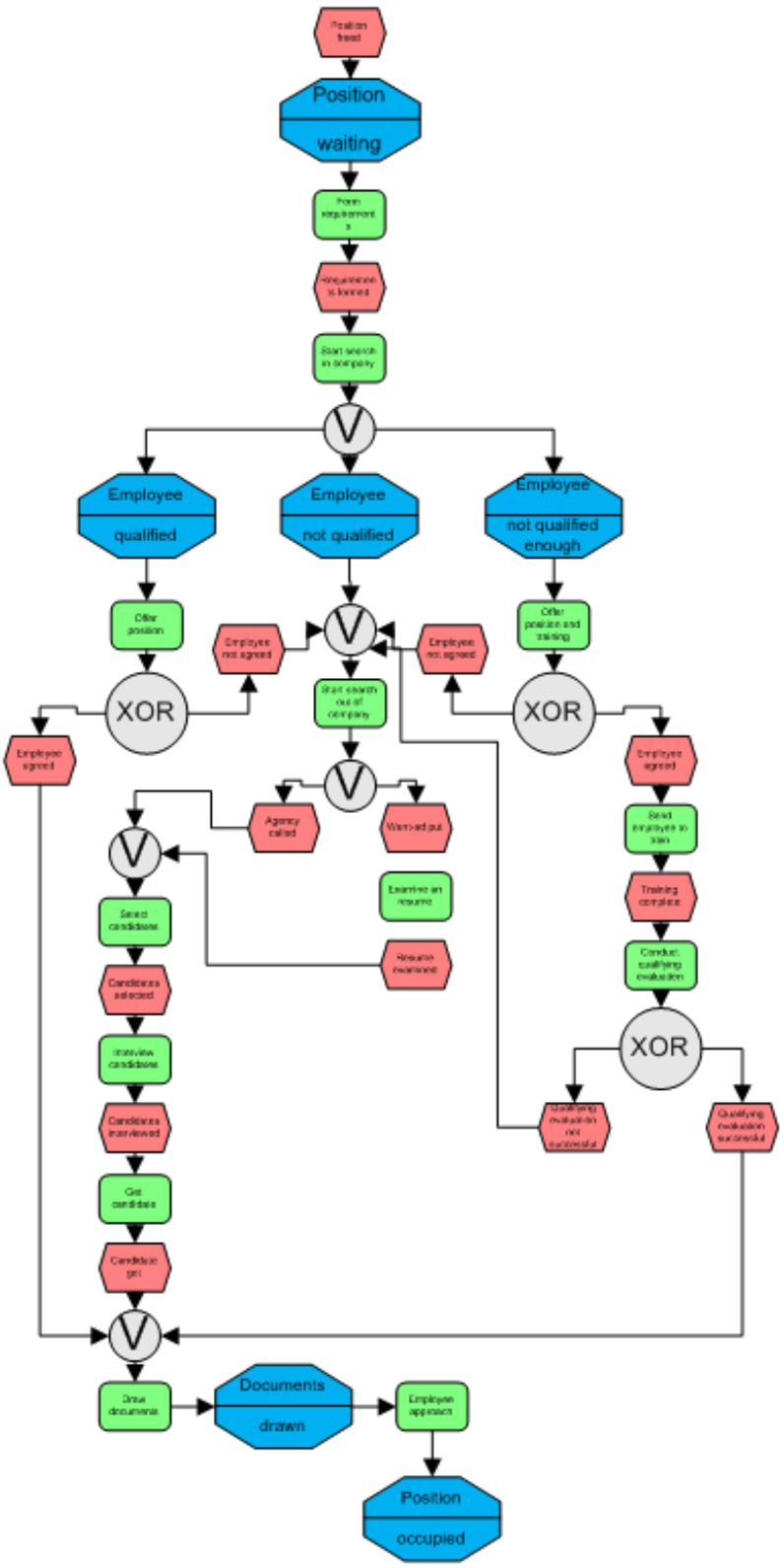

Figure 4. **Integrated Notation**

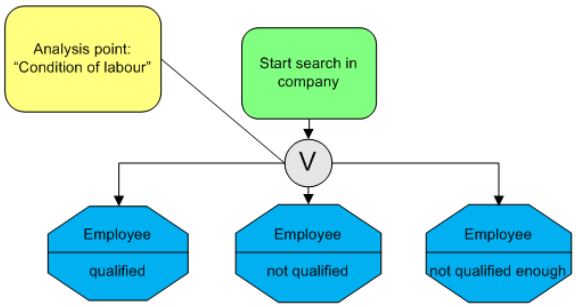

Figure 5. **Analysis Point**

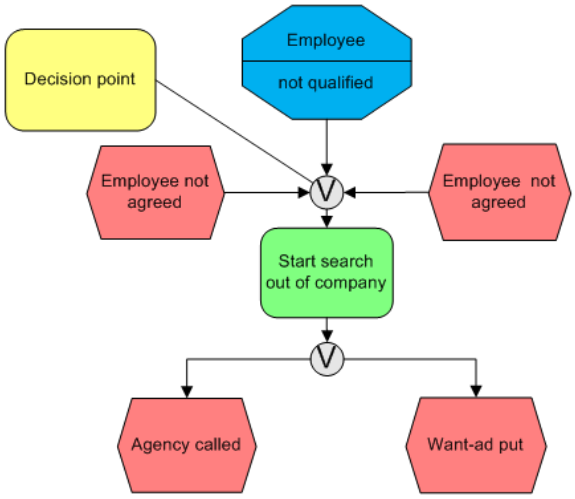

Figure 6. **Decision Point**

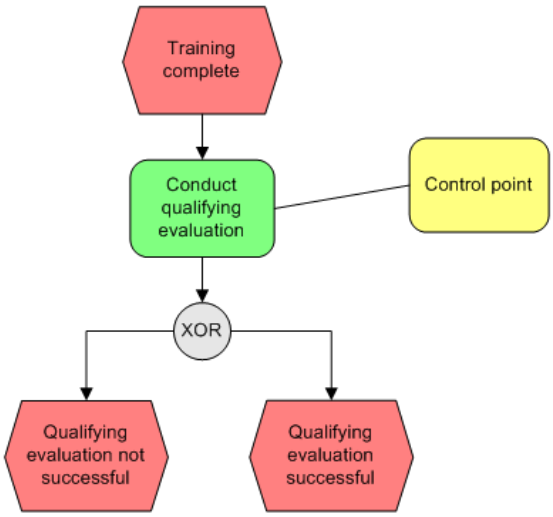

Figure 7. **Control Point**

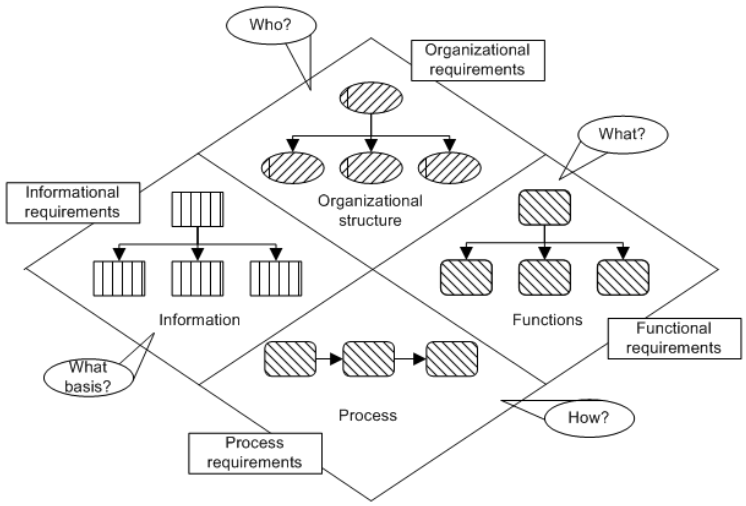

Figure 8. **Diagrams of ARIS**

Thereby the modeling environment should maintain the formal level of description of application domain apart from the conceptual level.

For instance taking into consideration the example above the following rules should be determined in Tab. 1.

Table 1

Formal description of rules

Rules	**Formal description**
If there is a worker, whose characteristics are the same as required ones, than we should propose freed position to him	$\forall a \in En_{ORG:POS} : (Co(a, freed) \cup (\exists b \in En_{RES:HUM}): EQ(F(a), F(b))) \rightarrow Ac(take, b, a)$
If there is a worker, whose characteristics are almost the same as required ones, than we should propose freed position to him	$\forall a \in En_{ORG:POS}\, (Co(a, freed) \cup \exists b \in En_{RES:HUM} : (F(b) \subset H(a))) \rightarrow Ac(training, b)$
If there are no workers with characteristics to compete with required ones, than we should start searching outside the company	$\forall En_{ORG:POS} : Co(a, freed)\ \nexists b \in En_{RES:HUM} : (F(b) \subset H(a)) \rightarrow Pr(startSearching)$

Those rules represent the analysis point, where the subject of analysis is the condition of the labour. Than the formal representation of search process can be shown as follows below:

$$\forall x \in En_{ORG:POS} : \exists y \in En_{RES:HUM} :$$
$$\{(Co(x, freed) \rightarrow Co(x, \text{waiting})) \rightarrow Ac(performFunctions, y, G(x))\} \rightarrow$$
$$\rightarrow [\{[P_1(y, \text{atCompany}) \wedge P_2(y, H(x))] \rightarrow Ac(\text{offer}, y, x)\} \wedge$$
$$\wedge \{[(P_1(\text{jobOffer}, \text{signed}) \rightarrow Ac(\text{draw}, y, En_{DOC}(\text{contract}))) \rightarrow$$
$$\rightarrow Co(x, \text{occupied})] \vee [P_1(\text{jobOffer}, \text{unSigned}) \rightarrow Ac(\text{startSearch}, y, y)]\} \vee$$
$$\vee \{P_1(y, \text{notAtCompany}) \rightarrow \text{Pr}(\text{startSearch})\}]$$

$$\Pr(\text{startSearch}) \to ((P_3(f_i(x), \text{top}) \to Ac(\text{callForAgency})) \to$$
$$\to Ac(\text{getCandidate}, En_{RES:HUM}(\text{recruiter}), En_{RES:HUM}(\text{futureEmployee}))\wedge$$
$$\wedge \Pr(\text{draw}) \wedge Co(x, \text{occupied}) \vee ((P_3(f_i(x), \text{common}) \to$$
$$\to Ac(\text{leaveAds}, En_{RES:HUM}(\text{recruiter}), En_{DOC}(\text{Ads}))) \to$$
$$\to Ac(\text{getResumes}, En_{RES:HUM}(\text{recruiter}))\wedge$$
$$\wedge Ac(\text{examineResumes}, En_{RES:HUM}(\text{recruiter}), En_{DOC}(\text{resumes}))\wedge$$
$$\wedge Ac(\text{interviewCandidates}, En_{RES:HUM}(\text{recruiter}), En_{RES:HUM}(\text{candidates}))\wedge$$
$$\wedge Ac(\text{getCandidate}, En_{RES:HUM}(\text{recruiter}), En_{RES:HUM}(\text{futureEmployee}))\wedge$$
$$\wedge \Pr(\text{draw}) \wedge Co(x, \text{occupied})$$

The formal level of business process description contains:

- The definition of sorts of variables: $En_{ORG:POS}$, $En_{RES:HUM}$, En_{DOC} — a spesific set of organization units, employees and documents. $En_{RES:HUM}(x)$ — not connected varibles of the sort.
- The description of the predicates below: $Ev(name)$ — an event with "name", $Co(object, name)$ — a condition with the *"name"* of *an "object"*, $Ac(name, [object], [subject])$ — an action with a *"name"* by *the "object"* at *"subject"*, $Pr(name)$ — a process with a *"name"*, $P_1(x, y)$ — an object "x" is in condition "y", $P_2(x, y)$ — an object "x" obtains characteristic "y", $P_3(x, y)$ — an object "x" conforms to an object "y".
- The definition of the functions below: $F(x) = \{f_1(x), \ldots, f_n(x)\}$ — a specific set of characteristics of the object "x", $G(x) = \{g_1(x), \ldots, g_n(x)\}$ — a specific set of distributed functions to an organization unit with the "x" name $H(x)$ — a specific set of minimal object "x" characteristics [4].

The following rules are aimed to define the completeness and integrity of the application domain (Tab. 2).

Table 2

The estimation of completeness and integrity example

Rules	**Formal description**
If the condition of a position is "waiting" its functions must be distributed	$\forall a \in En_{ORG:POS} : \exists b \in En_{RES:HUM}\ Co(a, \text{waiting}) \to Ac(performFunctions, b, G(a))$
The model is comprehensive if every organization unit has a model of distribution	$\forall a \in En_{ORG} : existsb \in En_{RES:HUM} : Ac(performFunctions, b, G(a)) \to$ complete
The estimation of completeness and integrity equals to the estimation of function's distribution	$\exists a \in En_{ORG:POS}\ \exists b_1, b_2 \in En_{RES:HUM} : \{Co(a, freed) \to Ac(performFunctions, b_1, G(a))\} \cap \{Co(a, freed) \to Ac(performFunctions, b_2, G(a))\} \to$ not integral

6. Conclusion

Nowadays there are a lot of ways of modeling and analyzing corporation activity and its management system, but special methodologies and modeling languages are limited in number, covering only one view in the description chosen. Modeling tool environments haven't reached their global optimum yet and their development and perfection should be continued. This process of improvement concerns both technologies and aids of application domain formal level description. Thus modeling environment must maintain not only conceptual level, but the formal level of application domain description, which contains model's characteristics, as well.

The article touches upon the direction of enterprise modeling instrument development. There has been shown an example of integrated notation, extending EPC diagram by adding new element "condition". Also the article contains the instance of so called formal level usage to estimate the completeness of the application domain description.

References

1. GOST 34.320-96: Information technology. Database standards system. Concepts and terminology for the conceptual schema and the information base. — 2003.
2. Моделирование бизнеса. Методология ARIS: Практическое руководство / А. Шматалюк, М. Ферапонтов, А. Громов, М. Каменова. — М.: Вест-Мета-Технология, 2003. — 333 с. [Modelirovanie biznesa. Metodologiya ARIS: Prakticheskoe rukovodstvo / A. Shmatalyuk, M. Ferapontov, A. Gromov, M. Kamenova. — M.: Vest-Meta-Tekhnologiya, 2003. — 333 s.]
3. *Booch G., Rumbaugh J., Jacobson I.* The Unified Modeling Language Usere Guide. — New York: Addison-Wesley Professional, 2005. — 496 p.
4. *Верещагин Н. К., Шень А.* Лекции по математической логике и теории алгоритмов. Языки и исчисления. — М.: МЦНМО, 2000. — 291 с. [*Verethagin N. K., Shenj A.* Lekcii po matematicheskoyj logike i teorii algoritmov. Yazihki i ischisleniya. — M.: MCNMO, 2000. — 291 s.]

УДК 681.518.2+658.511

Развитие инструментов моделирования предприятия для корпоративных систем управления

Г. М. Новикова, Т. В. Малютина

Кафедра информационных технологий
Российский университет дружбы народов
ул. Миклухо-Маклая, 6, г. Москва, Россия, 117198

Рассмотрены недостатки и обозначены направления развития инструментальной среды моделирования корпоративных систем управления, приведен пример интегрированной нотации, включающей понятия события и состояния, показано использование формального уровня описания предметной области для оценки полноты и непротиворечивости моделей.

Ключевые слова: моделирование, ARIS, корпоративные системы управления, интегрированная нотация, логические модели, формальные модели.