ISSN (e): 2250 – 3005 || Vol, 04 || Issue, 4 || April – 2014 ||
International Journal of Computational Engineering Research (IJCER)

Development of the theoretical bases of logical domain modeling of a complex software system

Oleksandr Dorensky[1], Alexey Smirnov[2]
1 Lecturer, Department of Software, Kirovohrad National Technical University, Kirovohrad, Ukraine
2 Professor, Department of Software, Kirovohrad National Technical University, Kirovohrad, Ukraine

ABSTRACT
Software of modern information and telecommunications systems, as well as of automated control systems is characterized by complexity. Therefore, in the initial stages of the life cycle of complex software systems there exists an actual problem of the application of effective technologies decomposition and logical domain modeling, as well as of software design. The paper proposes a mathematical formulation of the definition of the basic conceptual units of logical domain model for the creation of complex software systems. The proposed mathematical formalization of the logical structure of the domain model in terms of objects, relationships, attributes and states is the theoretical basis of mathematical formalization of techniques of logical domain modeling using object-oriented technology.

Keywords: *software, object-oriented technology, modeling, data domain, logical model, object, link, attribute, condition.*

I. INTRODUCTION

Currently, information systems and technologies allow providing the automation of practically all the areas of human activity. This has become a consequence of the need to process considerably greater amounts of accumulated information and complexity of managing technologies of complex systems that ensure the functioning of many industries and activities of mankind. One of the main factors underlying this process is the rapid development of digital and computer technology, universal informatization and globalization of society. Thus, the development and implementation of complex software systems is one of the most important tasks of our time.

Modern information systems and automated control systems are characterized by being multicomponent, having many interactions of functional elements, processing and exchange of large amounts of data, having elements of competition while using of the system resources and, consequently, by the complexity of the design and development of software. The complexity of software systems is caused by the complexity of the real domain, the difficulties of managing the development process, the need to provide sufficient flexibility to a program, as well as by an unsatisfactory ways of behavior of large discrete systems [1]. Thus, currently, a scientific and technical challenge of designing, developing new and improving existing software complex systems is important.

Software implementation of any complex system requires the use of the domain decomposition, which is represented as its partition into constituent elements. Thus, the domain refers to a part of the real world, which is a medium definition and implementation of a specific automated process or a group of processes [2].

The domain decomposition is performed by one of the common schemes: structural (algorithmic) decomposition and object-oriented decomposition [3]. The basis of the first scheme is the partition by actions (algorithms) and is used in the development of simple software. Structural technology of the software considers the entire system as the function, which is parted into sub-functions (procedures), then on sub-sub-functions etc. Object-oriented decomposition provides decomposition into autonomous objects. Thus, in object-oriented technology, unlike in the structural, the basic unit is not a function, but a class of objects, each of which consists of methods (actions) and the data [3.5].

Application of object-oriented technology in the stages of analysis and design of complex software systems (CSS) allows offering a wide range of logic models, based on a unified notation. This approach to programming and maintenance stages is developed in detail in the form of object-oriented programming concept, which is displayed in the topology of modern languages and integrated systems [6-8].

However, the effective application of technology of creation of complex software systems (automated control systems, information and telecommunication systems, etc.) on an industrial scale may be achieved subject to the development of software tools software of supporting technological methodology development. The creation of such a software tools determines nontrivial tasks reflecting the objectively existing contradiction between the high level of the declarative language used by a person in the process of software development on the one hand, and the need for low-level of language of machine realization of the programs on the other hand.

One of the ways to solve this contradiction is a mathematical formalization of the logical methods of modeling the domain of complex software systems.

II. MATHEMATICAL FORMALIZATION OF THE DEFINITION OF BASIC CONCEPTUAL UNITS OF LOGICAL DOMAIN MODEL FOR CREATING SOFTWARE

On a conceptual level, the domain is represented by a logical model that describes the key abstractions of the domain. The basic conceptual units of a logical model are objects [1, 2, 4].

Object is an abstraction of a set of elements of the domain that are related by common structure and behavior [3, 4]. As elements of the domain can act as real world objects, aims and destination of the objects, incidents in the domain, the relationship between objects that have the dynamics of behavior, rules or standards specified in the domain, etc. [3, 5, 9].

Because the logical domain model (LDM) is the image of a part of the real world, the definition of the basic abstractions of a logic model is a gnoseological process based on dialectical categories of unitary, special and general.

The category of the unitary displays the domain elements that are given to the analyst in the feeling. Feeling is the first stage of learning, reflection of individual properties, features, parties of a domain. Lets denote the set of such properties allocated by the analyst in the process of cognition, P_Ω. These properties are the properties of the domain elements allocated by the analyst, which we denote as Ω. Thus, the result of the feeling is also the function of identification of the domain

$$\chi_p : \Omega \times P_\Omega \to \{0,1\},$$

where

$$\chi_p(\omega, p) = \begin{cases} 1, \text{ if the condition is performed } p-\text{element property } \omega, \\ 0, \text{ if the condition is not performed }. \end{cases} \tag{1}$$

The next stage of learning is the representation by the analyst of a category of special as playing in the minds previously perceived properties of features, sides of the domain P_Ω. This process is abstracting of the important, from the viewpoint of the analyst, characteristics of the domain elements, the set of which lets denote as C. The result of a presentation of the category of special may be formalized as a surjective representation $F_p : P_\Omega \to C$, which defines a partition of a set $P_\Omega\{P^C\},\ c \in C$, where

$$\forall p \in P^C \ \ F_P(P) = c\,; \tag{2}$$

$$\forall p \notin P^C \ \ F_P(P) \neq c\,. \tag{3}$$

The next stage of learning is a synthesis based on the analysis of the unitary and, especially, general – of LDM objects. Based on the dialectical relationship of a unitary, specific and general, the process of such a synthesis may be formalized as an injective reflection $F_O : O \to \sigma(\Omega)$, where $\sigma(\Omega)$ - a set of all subsets of a set Ω, O – a set of objects of LDM, thus, that

$$[F_O(o) = \Omega_o] \Leftrightarrow [\exists \gamma \subseteq C((\forall \omega \in \Omega_o \forall c \in \gamma(\forall p \in P^c \chi_p(\omega, p) = 1))\,\& \\ (\forall \omega \in \Omega_o \exists c \in \gamma(\forall p \in P^c \chi_p(\omega, p) = 0)))], \tag{4}$$

$$\bigcup_{o \in O} F_O(o) = \Omega\,. \tag{5}$$

A subset Ω_o of a set of elements of the domain is abstracted as an object o if and only if there exists a set of abstractions of characteristics of the elements of the domain γ, common to all elements of the set Ω_o, and for each element of the domain that does not belong to the set Ω_o, in the set γ there is an abstraction of the characteristic which is not inherent to this element. Each element of the domain is included in a set, abstracted as an object of LDM.

Display F_O (formulas 4-5) defines a partial order on the set O by the following:

$$[o' \leq o''] \Leftrightarrow [F_O(o') \supseteq F_O(o'')]. \tag{6}$$

The next important conceptual units of LDM are communications. Communication is an abstraction of the set of relationships that systematically occur between different elements of the domain [1, 2, 4]. That the element of the domain is associated by a certain relationship with another element of the domain, and must be displayed on the stage of presentation of the category of special by an abstraction of the characteristics depending on another element. Thus, we define a certain set of dependencies D, where $D \subset C$. Lets define the set of connections of LDM through R. Then, if we define a surjective display $F_R : D \to R$ so, that

$$D^1 = \bigcup_{F_R(d)=r} \{d\}, \tag{7}$$

is based on the definition of communication [1, 4, 9]

$$\forall r \in R \left|D^r\right| = 2, \tag{8}$$

$$\begin{gathered} \forall r' \in R, \\ \forall r'' \in R(r' \neq r'') \Rightarrow \\ \left(D^{r'} \cap D^{r''} = \varnothing\right). \end{gathered} \tag{9}$$

Thus, abstracting by the analyst of communication r between the objects o' and o'' the LDM may be formalized as the statement:

$$\begin{gathered} \exists d' \in D \exists d'' \in D(d' \neq d'') \& \\ (F_R(d') = F_R(d'') = r) \& \\ \left(\forall \omega \in F_O(o') \exists p \in P^{d'} \chi_p(\omega, p) = 1\right) \& \\ \left(\forall \omega \in F_O(o'') \exists p \in P^{d''} \chi_p(\omega, p) = 1\right). \end{gathered} \tag{10}$$

Theorem 1. If o', o'', o'^*, o''^* belong to the set as the object of LDM, and there if a connection $r \in R$ exists between the objects o' and o'', then from the conditions $o' \leq o'^*$ and $o'' \leq o''^*$ there follows that relationship r exists between the objects o'^* and o''^*.

Proof. A proof of the theorem is trivial and follows the determination of the relationship (formula 10) and definition of a partial order on the set O (formula 6). At the stage of submission of the category of special two kinds of conceptual units of LDM are defined: state and attributes [1, 2, 4].

State is an abstraction of a characteristic of the element's position in its life cycle, which uses a specific set of rules, behavior patterns, and physical requirements and physical law. Attribute is an abstraction of a certain characteristic, which have all the elements abstracted as an object domain, which is not a dependence or condition [1, 4, 9]. Let's define through A a set of attributes of LDM, through S - a set of states of LDM. Thus,

$$C = A \cup S \cup D; \tag{11}$$

$$A \cap S = \varnothing; \tag{12}$$

$$A \cap D = \varnothing; \tag{13}$$

$$S \cap D = \varnothing. \tag{14}$$

Lemma 1. Let's define Z_o as the set of such $\gamma \subset C$, for which the following is performed:

$$\begin{gathered} \left(\forall \omega \in \Omega_o \forall c \in \gamma\left(\exists p \in P^c \chi_p(\omega, p) = 1\right)\right) \& \\ \left(\forall \omega \notin \Omega_o \forall c \in \gamma\left(\exists p \in P^c \chi_p(\omega, p) = 1\right)\right). \end{gathered} \tag{15}$$

Then Z_o forms a semilattice under the operation of the sets merge.

Proof. Since the merge operation has the properties of associative, commutative and idempotent, to prove the lemma we must prove that for any γ_1 and γ_2, which belong to Z_o, the condition $\gamma_1 \cup \gamma_2 \in Z_o$ is fulfilled:

$$\begin{gathered} [\gamma_1 \in Z_o \,\&\, \gamma_2 \in Z_o] \overset{(15)}{\Rightarrow} \\ \left[\forall \omega \in \Omega_o \left(\exists c \in \gamma_1 \left(\forall p \in P^c \chi_p(\omega, p) = 1\right)\right) \& \right. \\ \left(\exists c \in \gamma_2 \left(\forall p \in P^c \chi_p(\omega, p) = 1\right)\right) \& \end{gathered}$$

$$\left(\forall\omega\notin\Omega_o\left(\exists c'\in\gamma_1\left(\forall p\in P^{c'}\chi_p(\omega,p)=0\right)\right)\&\right.$$
$$\left.\left(\exists c''\in\gamma_2\left(\forall p\in P^{c''}\chi_p(\omega,p)=0\right)\right)\right)\Big]\Rightarrow$$
$$\Big[\left(\forall\omega\in\Omega_o\forall c\in\gamma_1\cup\gamma_2\left(\exists p\in P^{c}\chi_p(\omega,p)=1\right)\right)\&$$
$$\left(\forall\omega\notin\Omega_o\forall c\in\gamma_1\cup\gamma_2\left(\forall p\in P^{c}\chi_p(\omega,p)=0\right)\right)\Big]\overset{(15)}{\Rightarrow}$$
$$\left[\gamma_1\cup\gamma_2\in Z_o\right].$$

The lemma is proved.

Consequence. For any object of LDM o there exists the largest element in the set Z_o

$$C_o=\max_{\gamma\in Z_o}\gamma, \tag{16}$$

if the order is defined as follows:

$$\left[\gamma_i\le\gamma_j\right]\Leftrightarrow\left[\gamma_i\cup\gamma_j=\gamma_j\right]. \tag{17}$$

The assertion of Lemma 1 allows determining the sets of attributes and states by the object of LDM as an injective display $F_A:O\to\sigma(A)$ and $F_S:O\to\sigma(S)$ where $\sigma(A)$ is the set of all subsets A, $\sigma(S)$ - the set of all subsets S, as follows:

$$F_A(o)=A\cap C_o, \tag{18}$$
$$F_S(o)=S\cap C_o. \tag{19}$$

Lemma 2. Combining the range of the display F_O – $\Phi_O=\{F_O(o)\}$, $o\in O$, - and the empty set $\varnothing$ forms a semilattice under the operation of the intersection of sets.

Proof. Since the operation of the intersection has the properties of associativity, idempotency and communicativeness, to prove the lemma we must prove that for any Ω' and Ω'', belonging to Φ_O, the condition $\Omega'\cap\Omega''\in\Phi_O$ or $\Omega'\cap\Omega''=\varnothing$ is fulfilled:

$$\left[\Omega'\in\Phi_O\ \&\ \Omega''\in\Phi_O\right]\overset{(4,16)}{\Rightarrow}$$
$$\Big[\exists o'=F_O^{-1}(\Omega)\Big(\left(\forall\omega\in\Omega'\forall c\in C_{o'}\left(\exists p\in P^{c}\chi_p(\omega,p)=1\right)\right)\&$$
$$\left(\forall\omega\notin\Omega'\forall c'\in C_{o'}\left(\exists p\in P^{c'}\chi_p(\omega,p)=0\right)\right)\Big)\&$$
$$\exists o''=F_O^{-1}(\Omega'')\Big(\left(\forall\omega\in\Omega''\forall c\in C_{o''}\left(\exists p\in P^{c}\chi_p(\omega,p)=1\right)\right)\&$$
$$\left(\forall\omega\notin\Omega''\forall c''\in C_{o''}\left(\forall p\in P^{c''}\chi_p(\omega,p)=0\right)\right)\Big)\Big]\Rightarrow$$
$$\Big[\left(\Omega'\cap\Omega''=\varnothing\right)\text{ or }\Big(\exists o'=F_O^{-1}(\Omega')\exists o''=F_O^{-1}(\Omega'')$$
$$\Big(\left(\forall\omega\in\Omega'\cap\Omega''\forall c\in C_{o'}\cup C_{o''}\left(\exists p\in P^{c}\chi_p(\omega,p)=1\right)\right)\&$$
$$\left(\forall\omega\in\Omega'\cap\Omega''\forall c\in C_{o'}\cup C_{o''}\left(\exists p\in P^{c}\chi_p(\omega,p)=0\right)\right)\Big)\Big)\Big]\overset{(4)}{\Rightarrow}$$
$$\left[\left(\Omega'\cap\Omega''=\varnothing\right)\text{ or }\left(\Omega'\cap\Omega''=\Phi_O\right)\right].$$

The lemma is proved.

Lemma 3. If o' and o'' belong to the set of objects of LDM, then from the condition $o'\le o''$ it follows, that $C_{o'}\subseteq C_{o''}$.

Proof.

$$\left[o'\le o''\right]\overset{(6)}{\Rightarrow}\left[F_O(o')\supseteq F_O(o'')\right]\Rightarrow$$
$$F_O(o')\cap F_O(o'')=F_O(o''). \tag{20}$$

From the lemma 2 it follows: or $F_O(o')\cap F_O(o'')=\varnothing$, but this contradicts (20), or

$$\Big(\left(\forall\omega\in F_O(o')\cap F_O(o'')\forall c\in C_{o'}\cup C_{o''}\left(\exists p\in P^{c}\chi_p(\omega,p)=1\right)\right)\&$$
$$\left(\forall\omega\notin F_O(o')\cap F_O(o'')\exists c\in C_{o'}\cup C_{o''}\left(\forall p\in P^{c}\chi_p(\omega,p)=0\right)\right)\overset{(20)}{\Rightarrow}$$
$$\Big(\left(\forall\omega\in F_O(o'')\forall c\in C_{o'}\cup C_{o''}\left(\forall p\in P^{c}\chi_p(\omega,p)=1\right)\right)\&$$

$$\left(\forall \omega \notin F_O\left(o''\right) \exists c \in C_{o'} \cup C_{o''}\left(\forall p \in P^c \chi_p\left(\omega, p\right)=0\right)\right) \Rightarrow$$

$$C_{o''} \supseteq \left(C_{o'} \cup C_{o''}\right) \ \Rightarrow \ C_{o'} \subseteq C_{o''}.$$

The lemma is proved.

Theorem 2. If o' and o'' belong to the set of objects of LDM, then from the condition $o' \leq o''$ it follows, that $F_A\left(o'\right) \subseteq F_A\left(o''\right)$.

Proof. Following the formula (18),

$$F_A\left(o''\right) = A \cap C_{o''} \overset{\text{lemma } 3}{\Rightarrow}$$

$$A \cap C_{o'} \subseteq A \cap C_{o''} \Rightarrow F_A\left(o'\right) \subseteq F_A\left(o''\right).$$

The theorem is proved.

Theorem 3. If o' and o'' belong to the set of objects of LDM, then from the condition $o' \leq o''$ it follows, that $F_S\left(o'\right) \subseteq F_S\left(o''\right)$.

Proof. Following the formula (19),

$$F_S\left(o''\right) = S \cap C_{o''} \overset{\text{lemma } 3}{\Rightarrow}$$

$$S \cap C_{o'} \subseteq S \cap C_{o''} \Rightarrow F_S\left(o'\right) \subseteq F_S\left(o''\right).$$

The theorem is proved.

III. CONCLUSIONS

The proposed mathematical formalization of the structure of a logical model of the domain in terms of objects, relationships, attributes and states is the theoretical basis of mathematical formalization of techniques of logical domain modeling using an object-oriented approach. Formulated and proved assertions may be used to construct the algorithms for automated libraries of domain objects for developing object-oriented software of information and communication systems, as well as of automated control systems.

REFERENCES

[1] G. Booch. Object-Oriented Analysis and Design with Applications, 3rd edition. Addison-Wesley Professional publisher, 534 p., 2007.
[2] O.A. Smirnov , O.V. Kovalenko, Y.V. Meleshko. Software Engineering. EPL of KNTU, Kirovograd, 409 p., 2013.
[3] M. Glasser. Open Verification Methodology Cookbook. Springer, 290 p., 2009.
[4] S. Orlov. Technology creation software. Piter, SpB, 464 p., 2002.
[5] W.V. Siricharoen. Ontologies and Object Models in Object Oriented Software Engineering. IAENG International Journal of Computer Science, Volume 33, Issue 1, pp. 25-30, 2007.
[6] S. Swart, M. Cashman, P. Gustavson, J. Hollingworth. Borland C++ Builder Developer’s Guide. Sams Publishing, 1128 p., 2003.
[7] S. Teixeira, X. Pacheco. Borland Delphi Developer’s Guide. Sams Publishing, 1169 p., 2002.
[8] R. Lafore. Object-Oriented Programming in C++, Fourth Edition. Sams Publishing, 1012 p., 2002.
[9] A. Pillay. Object-Oriented Programming. School of Computer Science University of KwaZulu-Natal, Durban, 221 p., 2007.