International Journal of Engineering & Technology, 7 (4.5) (2018) 601-604

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Means to support the object traces in the conceptual modeling systems

L. Yu. Ismailova [1] *, S. V. Kosikov [2], V. E. Wolfengagen [1], J .V. Mazurova [3]

[1] National Research Nuclear University "MEPhI" (Moscow Engineering Physics Institute), Moscow, 115409 RF
[2] Institute for Contemporary Education "JurInfoR-MGU", Moscow, 119435 RF
[3] Federal State Budgetary Scientific Institution Scientific Research Institute of Eye Diseases, Moscow, 119021 RF
**Corresponding author E-mail: lyu.ismailova@gmail.com*

Abstract

The paper considers the problem of conceptual modeling of the domains, in which a large number of changes takes place, possesses a number of specific features. Some of the features are connected to the nature of the ongoing changes. The problem becomes severe when the domain contains a large quantity of independently acting agents that can produce changes in various parts of the model, the sets of interrelated data being changed by different agents independently.
Besides that, when supporting the dynamic nature models the possibilities of future changes in model objects might largely be deter- mined not only by their current state, but also by the past state - history of the objects in the model. The support to such history inside the current state of the object can be difficult, cumbersome or impractical for other reasons, which also stimulates the search for new ap- proaches to ensure the map of the changes of objects in the domain model.
It is offered to receive the decision by single out the individual information identities, that can be called state trajectories from the conceptual point of view and object traces from the implementation point of view. They represent a sequence of changes in the state of model objects essential for its support. The traces can fix not all changes of the corresponding objects, but only those which have an influence on the further behavior of the model. Various ways of supporting of traces are considered. The paper presents an approach to the determination and the support to the traces, based on using a combination of methods of the intensional logic and applicative computational systems.

Keywords: *Conceptual Modeling; Intentional Logic; Object Traces; Problem Domain; Temporal Logic*

1. Introduction

Conceptual modeling of the domains, in which a large number of changes takes place, possesses a number of specific features. Some of the features are connected to the nature of the ongoing changes. In those cases when a large number of objects change rapidly in the domain, the problem as it is becomes to support the processing of the information that displays the ongoing changes. This is a central problem when considering the conceptual model- ing systems from the point of "big data" views. The problem be- comes severe when the domain contains a large quantity of inde- pendently acting agents that can produce changes in various parts of the model, the sets of interrelated data being changed by differ- ent agents independently (including, asynchronously).
Another part of the specific features is associated with the need to preserve the integrity of the model when mapping the changes in the domain. When modeling the domain, rather complex constraints can be formulated, and they are overlaid on data values.
The constraints can bind not only the values relating to one state of the model, but also the data values in one or more previous system states. The support of such restrictions by means of the existing DBMS is sometimes difficult, which leads to the need to work out some specialized tools to support the integrity.
Besides that, when supporting the dynamic nature models the possibilities of future changes in model objects might largely be determined not only by their current state, but also by the past state - history of the objects in the model. The support to such history inside the current state of the object can be difficult, cum- bersome or impractical for other reasons, which also stimulates the search for new approaches to ensure the map of the changes of objects in the domain model.
As appeared a prospective approach to solving the problem is to single out the individual information entities that represent a sequence of changes in the state of model objects essential for its support. We call such entities state trajectories when consider it from the conceptual point of view and object traces when consider it from the implementation point of view. The trajectories can fixnot all changes of the corresponding objects, but only those, which have an influence on the further behavior of the model (including the needed ones to ensure the integrity).
The support to the trajectories can be performed for different objects in various ways. The selection of the support method for each object is carried out taking into account the requirements of the model as a whole. The paper presents an approach to the determination and the support to the object traces, based on using a combination of methods of the intensional logic and applicative computational systems.

2. Object traces

The refinement of the task for the support to the object traces requires a correct determination of state within the model. The point

Copyright © 2018 L. Yu. Ismailova et. al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

is that both the state of the object model and the model state generally (depending on the states of its component objects) can be defined in various ways. In particular, such determination depends essentially on the type of the information processing recognised in the model. This paper considers the computational models [1] that suggests the use of models for computing, including the adoption of computing model.

In case the computing model of von Neumann is adopted, as a rule, it is assumed that the model objects have attributes, the val- ues of which determine the state of the object. In the case of the applicative computational models [1], the state is determined in a little bit different way. The objects in this case have an applicative character. The values of the characteristics of the objects are de- termined as a result of application of a function to an argument. When calculating the function, the argument values are kept in the computing environment, which possesses the nested nature - ap- plicative objects in the most typical case are composed of other objects, and the object value calculation requires the computation of the values of some or all of the nested objects. The computation can be interrupted and then resumed. The computing environment may, thus, contain information representing partial computations and the one that is required for their restarting.

In the case of considering the abstract machine [1, 2] the state is represented by a set of machine registers. The values of the registers determine the course of the computation. Typically, the abstract machine may also switch between computing. In this case, one of the kinds of values that can be contained in the register is the data saved during the computation, which can be resumed later on.

The preceding consideration shows that the state of the model can be correlated with the computing environment if the model has computational character. The structure of the environment of computation determines the state of model, and hence the object trace. Such trace can be used to determine the computation carried out in the model, in particular, to resume the interrupted earlier computation or computation if their continuation is not required. It is also possible to extract the computations arguments out of traces and to process them, and also to check the constraints integrity.

The object traces can link to not only individual objects, but also to sets of model objects. Several model objects can have proper- ties that have to be changed in accordance with each other. Model- ing such changes requires the determination of states, integrating the properties of several objects in the set, and the further descrip- tion of the traces, reflecting the changes in the state of all the ob- jects introduced in the set.

Using the traces provides for the inspection and enforcement of the conditions imposed on the domain model. The conditions can be expressed either as usual constraints integrity or as properties of state trajectories. The computational character of the model thus allows to ensure not only examination of the properties of the motion along the state trajectory, but also the possibility of provid- ing conditions to achieve the desired properties.

3. Approaches to the support to the object traces

Using the concept of the state in modeling domains, in particular, the description of the model changes through the transitions from state to state is fairly traditional. This condition is interpreted in different ways depending on the nature and scope of the model application.

It seems that the most common interpretation of the condition is proposed in the framework of modal and intensional logic. The temporal logic [3] causes the greatest interest among the various known systems of modal logic to support the object traces. The temporal logic considers the dependence of the truth of language sentences on the time, and has the means to account such depen- dence in the formal toolkits. Thus, when constructing the models of temporal logic the language sentence is associated with a func- tion that assigns a certain value of truth to each moment of time. Such a function may be considered as one of the options of the trace.

Even more general interpretation is adopted in the intensional logic [4]. It generalizes the notion of time moment to the notion of the correlation point, which can be related to the set of time, spa- tial and other coordinates. Changing the state of the object in this case is modeled as a dependence of the characteristics values ob- ject on the correlation point. Such dependences are singled out as part of the formal toolkits of theory and are essentially used in the construction of interpretation. However, the proposed approach has a significant community, making it difficult to directly inter- face with systems of conceptual modeling and requires specifica- tion of the applied structures for their use in the framework of practical models. One of the methods of this specification will be discussed later (p. 5).

A more limited approach is associated with the definition of a system of states satisfying the applied limitation, and maintaining the system in such states. The trace in this case is defined so that it does not output the system beyond the admissible states. Within the DBMS, this approach is associated with a particular transac- tion, ensuring the implementation of constraints integrity [5]. This approach has proven its practical utility, but a set of constraints integrity must be defined by the database designer in advance, as a rule, without regard to the trace. In this case there is a risk of "los- ing" any constraint. Therefore, an approach based on the transac- tion appears to be unsuitable as the main support for the traces.

The possibility of movement along the object trace in both directions is also used in solutions retrieval systems, particularly in automated theorem proving systems. Inter alia the language Prolog, and others are widely known. One of the elements sup- porting the systems of such languages is the inverse operators, which are remembered when the system state changes. If the re- turn to one of the previous states is necessary, the inverse opera- tors are carried out, which restore this state.

In general, it appears that the presence of well-developed concepts and techniques that implement individual elements of support to the object traces, there is no common approach, ensuring the in- clusion of the concept of the trace into the conceptual modeling systems. For this reason the development of support tools to the object traces in the conceptual modeling of systems seems to be an actual task that has undoubted applied significance.

4. Task of the object traces support

The analysis of approaches to description of the states of the problem domain and the supporting of the traces of states allows for selecting promising ways of working with traces. Filing the comments regarding the relations between the object traces and computing environments allows for getting the computational nature to the concept of the trace, which provides the possibility of building supportive tools to the object traces

The preceding discussion allows to offer the setting of the support to the object traces task in conceptual modeling systems as a problem to develop the system of theoretical methods of description of the object traces, toolkits to support them and the methods of use. Theoretical methods of work with the object traces should provide:

- definition of the object state and description of the state support;
- determination of the possibility of the development of the state, depending on the parameter and description of such development (parameterization in time in this case serves as one of the ways of parameterization);
- description of the trace as an information entity, includ- ing the description of the properties of the traces;
- Possibilities of switching from the trace as an integral object to separate points on the trace, and vice versa.
- The toolkits of the object traces support should provide:
- possibility of supporting the object traces, defined by both the values at certain points, and description of change of state;
- carrying out computations "along" the object traces, in particular, the computation of requests for traces;

- Possibility of interpreting the trace as a specialized ele- ment of the computation environment.
- The methods of using the toolkits should provide:
- Ability to determine the properties of the object traces and the subsequent monitoring of properties;
- Implementation of the computations debugging along the object traces.

The three mentioned components, developed in accordance with the formulated requirements, constitute a subsystem of support to the object traces within conceptual modeling system. Special at- tention should be paid to the fact that the theoretical methods are an important integrated part of the system. As the practice of the work with the states shows (both in the field of intensional logic, and in the field of software systems design - See § 3), theoretically not properly thought through approach to the definition of the state results in vague and difficult-to-use systems. On the contrary, a strict definition of the state, a set of admissible states, etc., en- hances the expressive power of the relevant means and increases the reliability and safety of the resulting software systems.

5. Methods of support

After selecting the object traces become "objects of the first order" of the model. In particular, they in turn may be considered as objects, be analyzed, be involved in the computation and etc. To ensure the correct manipulation of objects of such type the use of developed formal methods is required [Ismailova].

The most natural method of theoretical description of the object traces seems to be the intensional logic. To provide the computational character of the model the integration of the intensional logic possibilities with the capabilities of the applicative computa- tional systems is desirable. Further on a version of the system, based on intensional logic and providing the basic abilities for description of the object traces is given.

5.1. System of intentional logic

When defining the intensional logic system it seems to be convenient to use the variant of designations [6]. The designations are based on the fact that the propositional connectives are considered as functions; thus, instead of p & q we write Kpq and etc. This method of designations is agreed with applicative point of view on the developed system. Note that when building the system the combinators are not used, this is why it is not possible to mix no- tation of system operations and combinators designations.

The alphabet systems comprise (1) the set of basic types e, t, s; (2) the set of variables $v_{i,\sigma}$ and constants and $a_{i,\sigma}$ for each type σ,
where i - a positive integer; variables will be marked as p, q, r, x, y, z, and the constants a, b, c; (3) propositional operations N, K, A, C, E; (4) quantified operation Π and Σ, λ - abstraction operation and intensional operations Δ and Θ; (5) intensional operators and L_i and M_i, where i - a positive integer; (6) brackets and comma.

The types will be ascribed to the expressions of the intensional logic language. Many types Type will be defined inductively by the following way: (i) e, t ε Type; (ii-1) if σ and τ ε Type, then (σ, τ) ε Type; (ii-2) if σ ε Type, then (s, σ) ε Type. Meaningfully we'll use the expressions like e to represent formalizations of domain objects, and t - for the presentation of statements about the do- main. The expressions of type (σ, τ) are functions from the type σ objects into objects of type τ. Expressions of type (s, σ) are ex- pressions of intensions type σ. The ratio ε is used for designation of the belonging to the metalanguage.

Let us define the set of terms Tm_σ for every type σ. $v_{i,\sigma} \in Tm_\sigma$; $a_{i,\sigma} \in Tm_\sigma$.

If $a \in Tm_{(\sigma, \tau)}$ ans $b \in Tm_\sigma$, then $(ab) \in Tm_\tau$.

If $a \in Tm_\tau$ and $v_{i,\sigma} \in Tm_\sigma$, then $(\lambda va) \in Tm_\tau$. If $a, b \in Tm_\sigma$, then $Eab \in Tm_t$.

If $a, b \in Tm_t$, then Na, Kab, Aab, Cab, M_iab, $L_iab \in Tm_t$. If $a \in Tm_t$ и $v_{i,\sigma} \in Tm_\sigma$, then Πva, $\Sigma va \in Tm_t$.

If $a \in Tm_\sigma$, then $\Delta a \in Tm_{(s, \sigma)}$. If $a \in Tm_{(s, \sigma)}$, then $\Theta a \in Tm_t$.

The terms of t type will also be called formulas. Note, that because the number of argument places for each method of definition of the term is fixed, there is no need for additional brackets (analysis of each term can be produced uniquely). Operators M_0ab and L_0ab will be given special meaning further on and they will be marked as Mab and Lab.

Now we define the semantics for the defined system. We define the set D, which will be considered as a set of individuals, and the set Asg, which will be regarded as the set of points of assignment [Ismailova, Kosikov]. Next we define the set of values for each type as follows.

$D_e = D$,

$D_T = \{0, 1\}$

$D_{(\sigma, \tau)} = D_\sigma \rightarrow D_\tau$, $D_{(s, \tau)} = Asg \rightarrow D_\tau$,

Where $A \rightarrow B$ – common designation for the functional space (set of functions from A into B).

We define a function F, which domain is the set of constants $a_{i,\sigma}$, and for every $a_{i,\sigma}$ of type σ is required $F(a_{i,\sigma}) \in D_{(s, \sigma)}$. The evaluation of the interpretation is performed in the environment ρ, which maps variable $v_{i,\sigma}$ to values, and for every $v_{i,\sigma}$ of type σ is required $\rho(v_{i,\sigma}) \in D_\sigma$. The interpretation is defined for every assignment point $k \in Asg$, i.e. is a function with the set of terms as domain, of type $\|\varphi\|\ \rho \in Asg \rightarrow D_\sigma$, where φ is a term of type σ and ρ is an environment. The interpretation is evaluated as fol- lows:

1) $\|a_{i,\sigma}\|\ \rho = F(a_{i,\sigma})$,
2) $\|v_{i,\sigma}\|\ \rho\ k = \rho(v_{i,\sigma})$,
3) $\|(ab)\|\ \rho\ k = \|a\|\ \rho\ (\|b\|\ \rho\ k)$,
4) $\|(\lambda va)\|\ \rho\ k = h$, where for $x \in D_\sigma$ is defined $h\ x = \|a\|\ \rho'\ k$, where $\rho'\ u = \rho\ u$, $\rho'\ v = x$;
5) $\|Eab\|\ \rho\ k = 1$ if and only if $\|a\|\ \rho\ k = \|b\|\ \rho\ k$;
6) $\|Na\|\ \rho\ k = 1$ if and only if $\|Na\|\ \rho\ k = 0$, and analo- gously for all propositional connectives;
7) $\|\Sigma va\|\ \rho\ k = 1$ if and only if there exists such $x \in D_\sigma$, that $\|a\|\ \rho'\ k = 1$, and analogously for Πva;
8) $\|\Delta a\|\ \rho\ k = \|a\|\ \rho$;
9) $\|\Theta a\|\ \rho\ k = (\|a\|\ \rho\ k)\ k$.

Formula φ is named true if $\|\varphi\|\ \rho\ k = 1$ and false otherwise.

5.2. Description of the object traces

The trace of the state in the proposed system is formalized as a relation on the set Asg. The intensional operator is the formal means to work with the object trace. The adopted method of speci- fying the semantics for intensional operators ensures, in particular,
the description of the changes of states of objects along the trace and definition of queries to objects taking into account their traces.

Let us consider a very simple example. Suppose that it is necessary to describe the object b, and its state does not change. For this the following formula can be used
MSbxΣyExΔy.

It can be verified by the direct evaluation. The value of the formula $\|MCbx\Sigma yEx\Delta y\|\ \rho\ k = 1$ if and only if for all k $\|Cbx\Sigma yEx\Delta y\|\ \rho\ k = 1$, if and only if for all k from bx follows that there exists such y, that in this assignment point k the value $\|x\|\ \rho\ k = \|y\|\ \rho$. The value of the right part does not depend on k, so it is obvious that the value of x is the same in all assignment points. So the state of the object really do not depend on the assignment point, i.e. re- mains constant.

6. Means of support

The support toolkits for supporting the conceptual modeling of domains, that contain the object traces, must combine general and specialized means of processing. The general means are unders- tood as the conceptual modeling tools that provide the mapping of domain objects, their properties and relations, as well as the changes

in the composition of objects and their interrelationships depending on the selected methods for the model parameteriza- tion. The specialized means are oriented to the work with the states and the traces, including the carrying out of computations along the trace.
For the purpose to exercise the definition techniques and the processing of the object traces the prototype editor of concept descriptions of domains was developed, it was implemented in Java language. The editor provides the support to conceptual mod- els in the form of sets of objects. Homogeneous objects can be grouped into classes. Objects, referred to the same class, are cha- racterized by a set of attributes that have typed values. The values may refer to one of the basic types or be a link to another object of the model.
For the purpose to exercise the techniques of tracking the object traces in models with different structure the editor ensures the possibility to extend the composition of the object classes and to change the sets of attributes, assigned to a certain class of objects (hereinafter - the extension). The extension is considered as one of the operations carrying out the movement along the object traces in the model, in connection with this the extension results can be processed by means of the editor intended to support the work with the traces.
The composition of the classes of objects, the set of attributes assigned to objects of a given class, the links between elements of different classes and other conceptual frameworks are fixed by means of describing in a specialized language, represented as XML dialect. The extension is implemented by changing the de- scription of the model structures with the following reinterpreta- tion of the description. This way ensures both the modification of the description of the conceptual structures, and the opportunity to use the external means for editing XML with the following import of the edited descriptions into the editor.
The set of object attributes with values, assigned to them, determines the state of the object. The change in the state may be carried out in an interactive mode of interaction with the user. The sequence of states, passed by the object, forms the trace of an object's state. The editor provides for storage of the traces, as well as the performance of a number of operations on it. The set of operations includes the search for the traces, as well as the com- parison of the traces for different objects. The comparison ensures the detection of the states, in which the objects have undergone various changes, and allows improving the nature of the identified differences.
The visualization of the processed conceptual structures presents some problems, because the composition of these structures (the set of attributes, links, types of attribute values, etc.) changes dur- ing the development of the conceptual model. For solving the
problems a special visualization language has been developed, which is an XML dialect, too. The language is based on the allocation of a number of standard visual components, providing the representation of the elements of an object and the ways of their change. The language defines a set of specific components used to visualize the objects of the model, and the way of their compositing in the dialog box, oriented to the user.

7. Conclusion

The present paper has proposed approaches to the identification and support of the object traces, based on the use of a combination of methods of the intensional logic and the applicative computa- tional systems. Seemingly, the proposed approaches provide for the following opportunities:

- setting the object traces for objects of conceptual model of domain;
- determining the properties of the object traces and veri- fication of their implementation;
- performing computations along the object traces;
- interpreting the traces as “objects of the first order” of the conceptual model;
- visualizing and modifying the individual elements of the traces;
- Debugging the conceptual model taking into account the traces.

The proposed methods and techniques were applied as methods of data processing when solving a number of problems from the field of law (modeling of the antimonopoly legislation, a number of issues of criminal law application, etc.). Building the models showed the practical applicability of the proposed techniques. Alongside with that, at this stage the models have private charac- ter, and the possibility of application of the object traces are not implemented up to the end. It looks like that the proposed tech- nique are amenable to development in the following areas:

- Possibility of taking into account the degree of com- pleteness of the description of the state;
- Integration with the means of the contextual description of the domain model;
- Ability to determine the specialized operations of the state manipulation.

The development of formalization methods for the object traces processing is expected to be promising to increase the power of conceptual modeling systems, integrity and security of supported models.

Acknowledgement

The research is supported in part by the RFBR grants 17-07-00893, 16-07-00914, 16-07-00892, 16-07-00912,
16-07-00909, 18-07-01082.

References

[1] Wolfengagen *VE, Methods and means of computation with objects*, M., JurInfoR, (2004).
[2] Wolfengagen VE, *Categorical Abstract Machine*, M., JurInfoR, (2002).
[3] Wolfengagen VE, Ismailova LY, Kosikov SV, “Computational Model of the Tangled Web”, *7th Annual International Conference on Biologically Inspired Cognitive Architectures*, BICA 2016, 16- 19 July New York, USA, Procedia Computer Science, Elsevier, Volume 88, 2016, pp. 306-311.
[4] Wolfengagen V, “Semantic Modeling: Computational Models of the Concepts”, *Proceedings of the 2010 International Conference on Computational Intelligence and Security (CIS '10), IEEE Com- puter Society, (2010)*, Washington, DC, USA, pp. 42-46.
[5] Ismailova LYu, Kosikov SV, Wolfengagen VE, “Applicative Methods of Interpretation of Graphically Oriented Conceptual Information*”, 7th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2016, 16-19 July New York, USA*, – Procedia Computer Science, Elsevier, Volume 88, (2016), pp. 341-346.
[6] Ismailova LYu, Kosikov SV, Wolfengagen VE, “A harmony and disharmony in mining of the migrating individuals”, 2016 Third In- ternational *Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)*, Moscow, (2016), pp. 52-57.