Problem-oriented Stratification of Transport Infrastructure Management

Tatiana Mikheeva
Institute of aviation technology
Samara University
Samara, Russia
mikheevati@gmail.com

Sergey Mikheev
Institute of aviation technology
Samara University
Samara, Russia
ms140-500@yandex.ru

***Abstract*—The paper analyses basic mathematical models on the basis of the fundamental laws of intellectual control. The models take into account of the sensitivity of the global systemic objective functional to the components of the set of states of the models in accordance with the systemic objectives definition principle. Decomposition of the underlying model is based on the information, methodological and functional principles of stratification. Based on stratification, the strata of the problem-oriented mathematical models are set up in the form of sets of private models and sequences of its mappings – morphisms. The morphisms meet the system quality criteria and are suitable for transport management. Stratified design based on patterned P-model, represented by the set of embedded spherical strata, mapping the strata of the domain space and relationship types between the space strata. Taxonomy allows you to design new types of objects based on old ones, inheriting their properties and methods.**

Keywords*—*stratification, taxonomic model, object, complex object

I. Introduction

The paper describes major aspects of system modeling, management and optimization of decision-making processes, synthesis of complexly organized structures of intelligent transport geographic information system, optimization and automation of transport processes when performing transport infrastructure management. Basing on the fundamental laws of intellectual control, we developed basic mathematical models taking into account sensitivity of the global systemic objective functional of the models to the components of the set of states of the models in accordance with the systemic objectives definition principle. Decomposition of the underlying model is based on the information, methodological and functional principles of stratification. We used both original methods, developed and described in the paper and known methods, among them the methods of the theory of traffic flow, as constructive methods. Based on stratification, the strata of the problem-oriented mathematical models are set up in the form of sets of private models and sequences of its mappings – morphisms. The morphisms meet the system quality criteria and are suitable for effective solution of optimal transport infrastructure management [1,2].

The models are synthesized on the basis of the following patterns:

- formalized description of decision support objectives;
- transport infrastructure management;
- investigation of intensity, density, composition of streams;
- traffic flow and traffic management facilities interactions;
- external environmental influences;
- technological, regime and design factors;
- constrained quality criteria for decision-making processes in transport infrastructure management.

Models are presented in the form of various mathematical constructions based on patterns: linear and nonlinear models, deterministic, fuzzy-defined and stochastic , distributed and zonal objects, analytical, neural network, geo-information, optimization models. To optimize the management of transport infrastructure we suggested using the methodology of a systemic taxonomic hierarchy of structures. The paper describes various aspects of the implementation of zonal management. We carried out its verification and obtained management quality assessment results. The paper describes the technique of semantic stratification of the transport infrastructure through the Osgood space. Stratification contains varieties of project strata. In the process of modelling the functioning of the transport infrastructure, the strata allow taking into account the fact that the resulting optimal solution makes sense not only due to its objective content, but also criteria related to the features of an intelligent plug-in. The plug-in makes decisions with regard to criteria, limits, external parameters influencing the outcome of the simulated process. While the decision process, modelling assessments from different strata are interdependent, interact with each over allowing us to determine the level of the most interacting scales and group them into factors [7,8].

II. Stratified Design. Base Stratification

Stratified design based on patterned P-model, represented by the set of embedded spherical strata, mapping the strata of the domain space and relationship types between the space strata. The strata are based on the static regulation of structured construction of classes of objects and composition of interclass relations [5].

The construction of program representations of any object specified by the taxonomic model is carried out on the basis of aggregation, where con_of means *«Consists of»*, con_in means *«Is a part of»; a, b, c* are class objects^

$$c\ con\_of\ (a,b) \Rightarrow a\ con\_in\ c\ \&\ b\ con\_in\ c\ .$$

Axiom ***1.*** Suppose $c\ is\_a\ C$ based on objects $a\ is\_a\ A$ and $b\ is\_a\ B$:

then

$$\forall c\ is\_a\ C \mid \tilde{P}^C\ C = \tilde{P}^C\ A \cup \tilde{P}^C\ B \Leftrightarrow$$
$$c\ con\_of\ a,b \mid a\ is\_a\ A\ \&\ b\ is\_a\ B$$

An object of a generic class is an aggregate $Agg(A*\rangle B)$, consisting of individual strata that are carriers of inherited and acquired properties. (Fig. 1).

The strata a_1, a_2 in objects b_1, b_2 are used as property carriers $\tilde{P}^C\ A$ (generalizing class *A*). The strata b'_1, b'_2 in the same objects are used as carriers of acquired properties $\tilde{P}'\ B = \tilde{P}^C\ B - \tilde{P}^C\ A$ (generalizing class *B*).

In the general case, the model of single inheritance *PSingleTaxon:*

$$A_1 *\rangle A_{12} *\rangle \ldots *\rangle A_n \qquad (1)$$

defines every object $a\ is\_a\ A_n$ as a stratified aggregate $a\ con\_of\ a_1, a'_2, \ldots, a'_n$, wherein $a_1\ is\_a\ A_1$ is the carrier of the basic properties in the object *a*; $a'_2, a'_3, \ldots, a'_n$ – the carriers of acquired properties. In this case, the set of strata layers $a_1, a'_2, a'_3, \ldots, a'_n$ in object *a* stores the full set of class properties *A*:

$$\tilde{P}^C\ A_1 \cup \tilde{P}^C\ A_2 - \tilde{P}^C\ A_1 \cup \tilde{P}^C\ A_3 - \tilde{P}^C\ A_2$$
$$\cup \ldots \cup \tilde{P}^C\ A_n - \tilde{P}^C\ A_{n-1} = \tilde{P}^C\ A_n \qquad (2)$$

For *PSingleTaxon* single inheritance models the construction of minimal objects, in which each property is represented by a single individual stratum, is typical. On the basis of the generalization relation, it is possible to construct non-minimal objects, in which each property can be represented by *n* different strata (storing generally different values of this property) [3,4].

Axiom ***2.*** Suppose $a\ is\_a\ A$ and $b\ is\_a\ B$ and $ab\ con\_of\ a,b$:

then $a\ is\_a\ A \Rightarrow a\ has\_a\ \tilde{P}^C\ A$;

$b\ is\_a\ A \Rightarrow b\ has\_a\ \tilde{P}^C\ A$;

$a\ has\_a\ \tilde{P}^C\ A\ \&\ b\ has\_a\ \tilde{P}^C\ A \Rightarrow$

$ab\ has\_a\ \tilde{P}^C\ A \cup \tilde{P}^C\ A \Rightarrow$

$ab\ has\_a\ \tilde{P}^C\ A \Rightarrow ab\ is\_a\ A.$

Any aggregate composed of objects of the same class is also an object of this class.

The generalization relation allows us to construct different objects using the same stratum, which is divided between these objects. Such a stratum not only determines the general property of objects, but also stores the general value of this property [6].

Constructing objects based on the multiple inheritance pattern *PMultipleTaxon* is associated with the creation of units, in which the number of strata exceeds the number of levels in the corresponding taxonomic model.

The complexity of objects created on the basis of taxonomic models is directly related to the level of the corresponding class in the tree. The base class contains the simplest objects, and the most complex contains the class that is the furthest descendant of the base. Thus, the regulation of taxonomic specification in the tasks of stratified construction of objects is associated only with the control of hierarchical relations in the tree and does not impose any restrictions on the choice of objects from which the aggregates are constructed.

Object design pattern mechanisms can be divided into two categories: *«default»* and *«based on»*. «Default» method is used only for the single inheritance of properties, in which each generalized class in a taxonomic tree is directly associated with only one generalizing class. This allows automatically creating minimal objects, as well as access to strata only on the basis of the taxonomic specification of the model. Design by *«based on»* principle requires explicitly specifying the object base in constructor operations. The use of this form of design allows significantly expanding the possibilities of a «pure» taxonomy by means of creating shared and non-minimal objects. This allows creating complex stratified structures. Setting a class with a set of immanent properties $\tilde{P}^C$ defines a potential set of class objects. At the same time, any class as an association of specific objects has group properties common to many class objects. An example of a group property is the number of

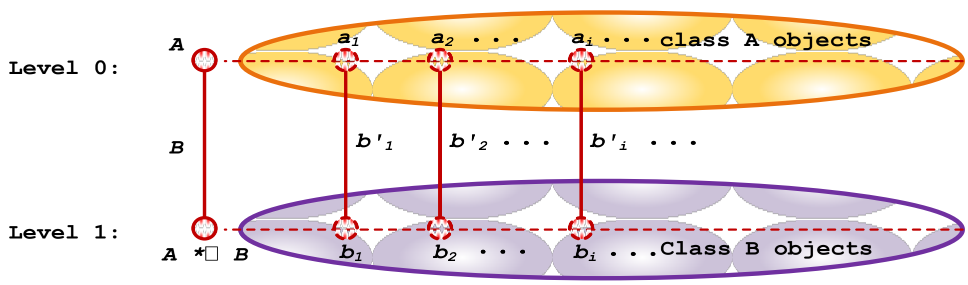

Fig 1. $B \langle * A$ object class structure

objects in a class. This property does not characterize objects in any way and, therefore, is not represented in the respective supports of immanent properties (object-oriented memory system). Group properties can be constructed on the basis of immanent. Such construction is associated with the selection of objects in a subclass. The definition of a subclass is carried out dynamically in the process of modeling the designed system; therefore, it cannot be represented in the static structure of a taxonomic tree.

Definition ***1.*** *Associative properties* as a generalization of group properties always have a value of many objects.

Any class has an associative property, the value of which is the set of objects of this class. The introduction of order relations on the set of objects - members of the association - allows for adequate design of dynamic structures that determine the development of simulation processes in the models of the systems under study. Among such structures are the reference information mode with the hypertext structure of the organization, chronologically ordered “event calendars”, ordered lists of priorities, table of contents of the book, etc.

Definition ***2.*** We define *Associative tree* as an ordered list structure, starting with *a Special Element* and consisting of elements, each of which is either an object or an association.

Definition ***3.*** As *a Special Element*, we use either a class of objects based on inheritance relations, or an association.

A class based on taxonomic relations defines a single Special Element of the associative tree, which is called the root and is the static vertex of the tree. Any association determines the “growth point” of the associative tree, in which a new branch can be dynamically created, which is also an associative tree. This recursive definition of an associative tree is illustrated by an example of the recursive structure of such a tree, shown in Fig. 2.

T-class defines a static vertex of an associative tree (root), A_1 , A_2, ... A_n – dynamic vertices. The dashed lines highlight associations; the remaining rectangles represent simple objects. The associations A_2 *u* A_3 in this figure are members of the association A_1 ; A_4, A_5 – members of the association A_2 ; A_3 – empty association; A_1 – a member of the association associated with the *T*-class. Any association is a member of a top-level association tree and contains its own set of members (possibly empty). Any dynamic operations on objects (creation-destruction, inclusion-exclusion from associations) do not change the overall organization of the associative tree. Moreover, any object can be a member of several associations only indirectly, for example, object *a* (Figure 1) is a member of associations T, A_1, A_2 : in A_2 it enters directly in A_1 through an intermediary and into *T*-class association through two intermediaries A_1 and A_2 . There are three members in the association A_2 , of which the first and second are associations A_4, A_5 , the rest are simple objects. The regular structure of an associative tree allows for the effective control of any dynamic modifications of the composition of its vertices.

III. Instrumental Decision Support Classes for Temporal Data Processing

The composition of instrumental classes of plugins supporting the chronological relationships of objects participating in dynamic processes includes the classes *Timer*, *Time*, which form the generic branch of a single inheritance. The described stratification of the immanent properties of these classes makes it possible to design various schemes for supporting chronological relationships [15].

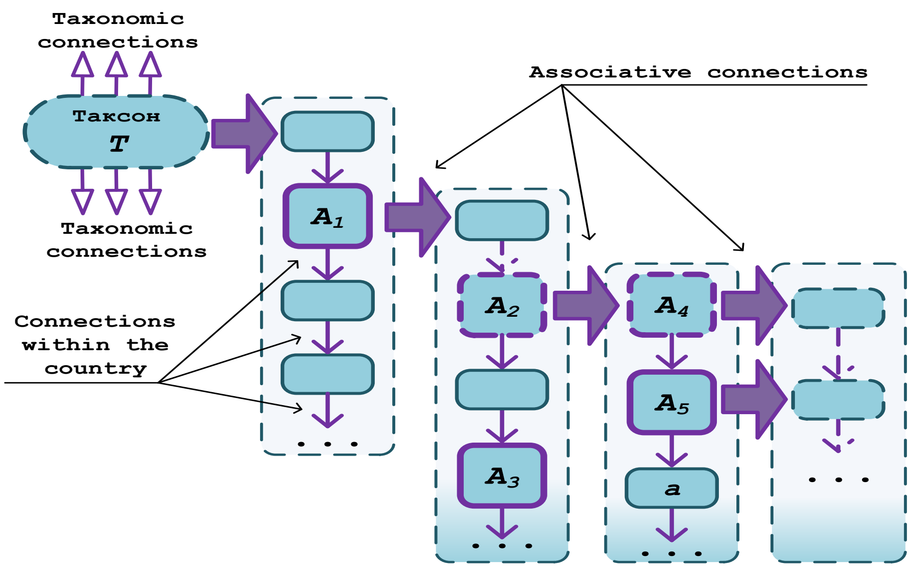

Fig 2 Associative tree

Temporal Data Pattern. Pattern ⇒ *PTimer:* objects are used as «*Timer*» (model time clocks). The structure of the time scale and its design methods are immanent properties of this class. An ordered set of units of time dimension (parts of the scale) determines the time scale, for example, HOUR: MIN: SEC (hours: minutes: seconds), YEAR: MONTH: DAY (year: month: days). Any time scale is determined by the length of the scale (the number of parts used (*Size_Scale*)) and methods of conversion from one dimension (part) to another (*MRecount*). So for the scale HOUR: MIN: SEC, such modules are determined by the values: $M_{1,2}=M_{sec,min}=60;$ $M_{2,3}=M_{min,hour}=60;$ for scale YEAR: MONTH: DAY: $M_{1,2}=28\div31;$ $M_{2,3}=12.$ Scale determination methods are methods for setting the values of the number of parts of the scale. (*Method* ⇒ *MSize_Scale*) and conversion modules (*Method* ⇒ *MRecount*).

Inherent properties of class objects «*Timer*»:

$$\tilde{P}^C \ \ Timer \ = (Scale, Size\_Scale, \\ Recount, MSize\_Scale, \\ MRecount \ ) \tag{3}$$

The «*Time*» class will be defined as a subclass of the «*Timer*» class: $Timer *\rangle Time$, the objects of this class $t \ is\_a \ Time \ A$ are points in time (timer readings, instants) at which changes in the state of the object of the subject area are possible. The immanent properties of a class $\tilde{P}^C \ Timer$ include the presentation of timer readings, methods for converting such readings from one timeline to another, methods for creating / destroying instant objects.

Temporal ***data visualization pattern.*** *Pattern* ⇒ *PShow_Time:* visualization of the timer reads determines the structure of the moment object, for example:

$$\forall time \ \big| \ time \ \ is\_a \ \ Time \Rightarrow time \ \ con\_of \\ (time_1 \ \big| \ time_1 \ \ is\_a \ \ Integer, \\ time_2 \ \big| \ time_2 \ \ is\_a \ \ Integer, \\ time_3 \ \big| \ time_3 \ \ is\_a \ \ Integer) \tag{4}$$

Such a representation is associated with the use of a timeline consisting of 3 parts, each of which is interpreted in accordance with the type *Integer*. The *Time class* defines the time algebra, – for this example, the set of triads $(time_1, time_2, time_3)$ and actions on them: determining the value (*MInit_Time*), methods for converting timer readings from one time scale to another (*MRecount_Time*), comparing (*MEq_Time*), converting to base machine types (*MTime_To_Real*) etc.

Any instant object is created based on a timer object, and multiple instants can be created based on a single timer (i.e., a timer object is a shared object).

Inherent properties of class objects «*Time*»:

$$\tilde{P}^C \ \ Time \ = (Timer, Show\_Time, \\ MInit\_Time, MEq\_Time, \\ MRecount\_Time, MTime\_To\_Real). \tag{5}$$

IV. Stratification of Transport Infrastructure. Taxonomic Stratification

Taxonomy allows you to design new types of objects based on old ones, inheriting their properties and methods. Fig.3 shows an example of a taxon-my tree describing a model of transport infrastructure (some of it). The semantics of classes are as follows.:

OS – object on the site;

SE – transport network section;

RS – road sign;

LS – light support;

CW – crosswalk;

CR – crossroad;

LPC – lighted pedestrian crossing;

CPC – crossroad with crosswalk.

The constructed taxonomic model is characterized by the presence of two generic classes (*OS, SE*), four single (*RS, LS, CW, CR*) heirs and two multiple (*LPC, CPC*) [14].

Any step in the process of constructing an object a is associated with the inclusion of the previous (generalizing) basis $\tilde{P}^C \ A_i$ in the form $\tilde{P}^C \ A_{i+1}$. The uniqueness of the object - the carrier of defining properties is that all its layers are individual strata. For example (Fig. 4), the construction of objects by specification $SE *\rangle CW, CR$ leads to the fact that in any object $cw \ is\_a \ SW$ as in any object $cr \ \ is\_a \ CR$ будет there will be its own stratum for storing properties $\tilde{P}^C \ SE$. Strata se_1, se_2 in objects cw_1, cr_1 are used as carriers of properties $\tilde{P}^C \ SE$ (generalizing class *SE*). Strata *cw'1, cpc'1* in the same objects are used as carriers of acquired properties:

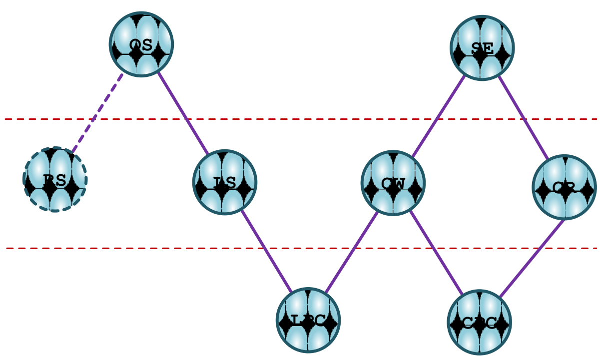

Fig. 3. Taxonomic tree model of transport infrastructure

$\tilde{P}^{C}{}'\ CW\ = \tilde{P}^{C}\ CW\ - \tilde{P}^{C}\ SE$ (generalizing class *CW*),

$\tilde{P}^{C}{}'\ CR\ = \tilde{P}^{C}\ CR\ - \tilde{P}^{C}\ SE$ (generalizing class *CR*);

$nn_1\ con\_of\ \ se_1, cw'_1\ ;$

$npк_1\ con\_of\ \ se_2, cr'_1\ .$

Designing objects based on multiple inheritance is associated with the creation of aggregates in which the number of strata exceeds the number of levels in the corresponding model (Fig. 5).

In this structure:

$lpc_1\ con\_of\ \ ls_1, cw_1, lpc'_1\ |$

$ls_1\ is\_a\ LS\ \&\ cw_1\ is\_a\ CW\ \&$

$\&\ lpc'_1\ is\_a\ LPC'/\tilde{P}^{C}{}'\ LPC\ ;$

$lpc_2\ con\_of\ \ ls_2, cw_2, lpc'_2\ |$

$ls_2\ is\_a\ LS\ \&\ cw_2\ is\_a\ CW\ \&$

$\&\ lpc'_2\ is\_a\ LPC'/\tilde{P}^{C}{}'\ LPC\ .$

The taxonomic model of unit-inheritance (Fig. 6) leads to the definition of non-minimal objects. The creation of such objects is connected with the introduction of additional rules determining access to strata. In the model of single-set inheritance, there are at least two different generalizing paths of the form $SE *\rangle\ ...\ *\rangle CPC$ [13].

By reducing the intermediate levels of generalization into one class, we get:

$$SE *\rangle CW, CR\ \&\ CW, CR *\rangle CPC\ . \tag{6}$$

Single inheritance $SE *\rangle CW, CR$ determines the objects of the *cw* and *cr*, in the structures of which there are various strata for storing generic properties $\tilde{P}^{C}\ SE$. Naturally, they will be included in the structure of class objects $CW, CR *\rangle CPC$ [12].

V. Stratification of Transport Infrastructure. Taxonomic Stratification

Managing a complex object requires the creation of a control object model. In a transport infrastructure management system, this is a traffic flow model.

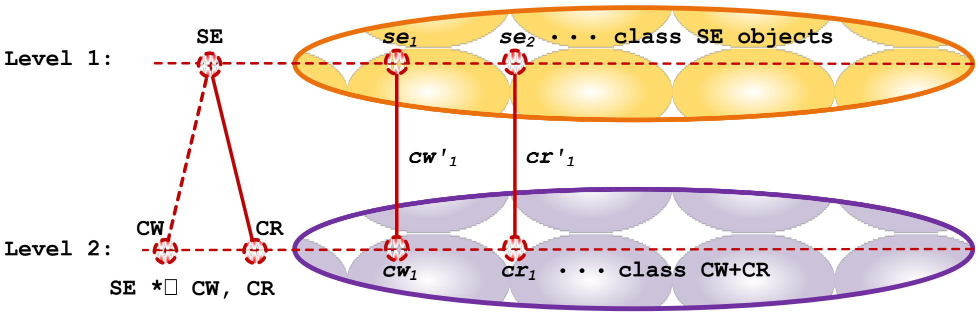

Fig 4. Constructing objects based on single inheritance

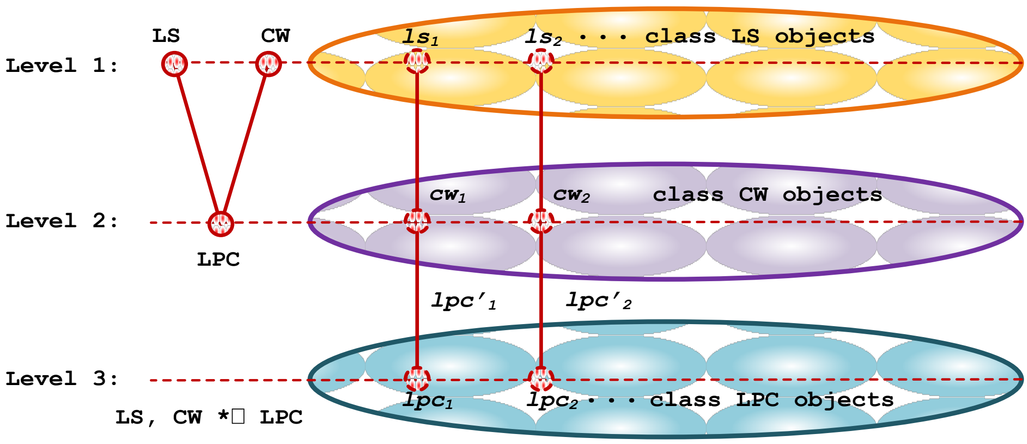

Fig. 5. Constructing objects based on multiple inheritance

The traffic flow $\tilde{S} = \{\tilde{s}_i\}$, as a control object, is a collection of a large number of discrete elements - cars.

Based on traffic studies and the practice of its organization, numerous gauges and criteria have been developed. Considered as a control object, a mesoscopic model of a transport stream $\tilde{S} = \{\tilde{s}_i\}$, moving along the arcs $\tilde{e}_i \in \tilde{E}$ of a graph of a transport network is characterized by the speed $v^{\tilde{e}_i}$, density $k^{\tilde{e}_i}$ and intensity $I^{\tilde{e}_i}$ of a stream on an arc $\tilde{e}_i$ at a time instant t_i.

Class «*Intensity*». Traffic intensity $I = \{\tilde{I}_k\}$, $k = 1, 2, ..., n$ – is the number of vehicles passing through the section of the road per unit time *t*. Of major importance in the problem of the organization of movement is the temporary unevenness of movement during the year, month, day, and even an hour. The class is specified by the following basic parameters:

- unique number of the intensity measurement result $number^I \in N = \{1, 2, ..., n\}$;
- arc Edge= $\tilde{e}_i$ graph G of the transport network Θ, on which the intensity is determined $\tilde{I}_k$;
- quantity in a stream of vehicles of a certain type $NVehicle \in Q^{1\times N}$, where $Q^{1\times N}$ – is a set of vectors of size N; traffic intensity of trams IntensTram = $\tilde{I}_i^T$ on a given section of the road (on the arc of the traffic network graph);
- pedestrian traffic intensity IntensPed = $\tilde{I}_i^P$ on a given section of the road (on the arc of the traffic network graph);
- date of measurement $Date\_On \in Date[YY : MM : DD]$; presentation format – YY: MM: DD;
- the start time of Begin_End_Time= $Time\_Begin \in Time\ [HH : MM : SS]$ and the end $Time\_End \in Time\ [HH : MM : SS]$ of the intensity measurement. The range of values – from 00 hours 00 min. till 23 hours 59 min.

The combination of these factors determines the immanent properties $\tilde{P}^C(I)$ of the class «Intensity»:

$$\tilde{P}^C(I) = (\tilde{e}_i, NVehicle, \tilde{I}_i^R, \tilde{I}_i^T, \tilde{I}_i^P, Time\_Begin, Time\_End, Date\_On).$$

An object of the «Intensity» class is constructed as follows:

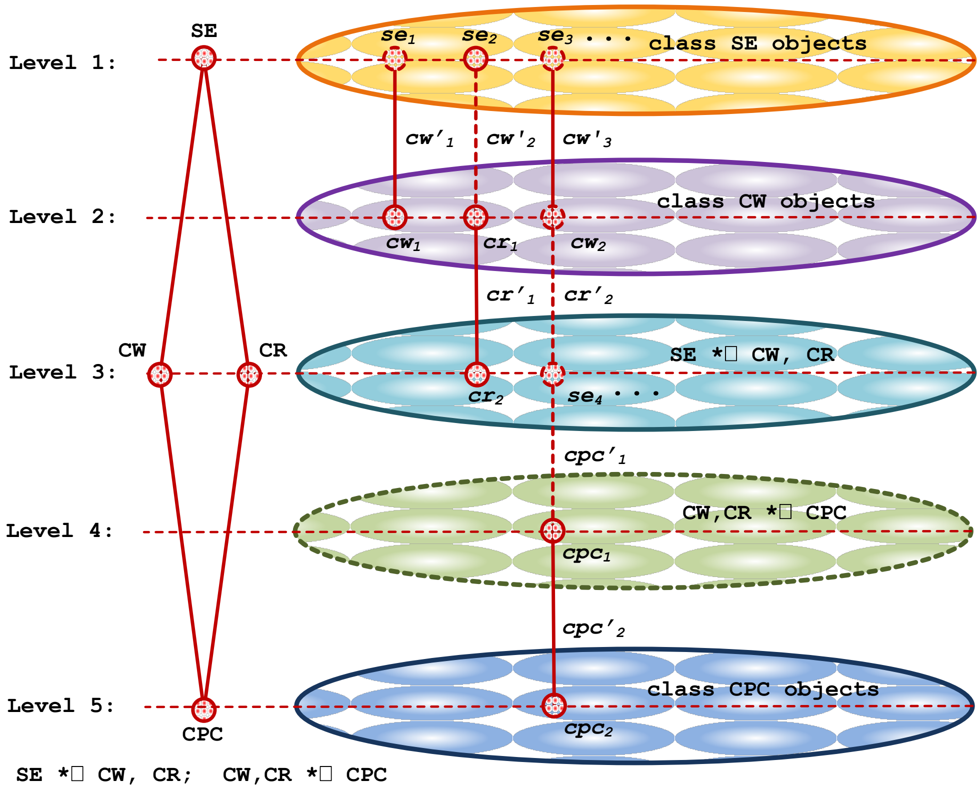

Fig. 6. Designing Objects Based on Single-Inheritance

$$
\begin{aligned}
\forall intensity \,|\; & intensity\ \ is\_a\ \ INTENSITY\ con\_of \\
& (Edge,\ Reduct\_Intens, Intens\_Tram, \\
& Intens\_Ped, Begin\_End\_Time, \\
& Intens\_Date)\,| \\
& (edge\ \ is\_a\ \ Edge), \\
& reduct\_intens\ \ is\_a\ \ Reduct\_Intens\ , \\
& intens\_tram\ \ is\_a\ \ Integer\ , \\
& (intens\_ped\ \ is\_a\ \ Integer), \\
& (begin\_end\_time\ \ is\_a\ \ Time), \\
& (intens\_date\ \ is\_a\ Time)
\end{aligned}
$$

The model of the «Intensity» class taking into account these constructions, will be determined by the following T-tree (Fig. 7). Taxonomic model of the class «Intensity» will look as follows:

$$
\begin{aligned}
& Time\ \ast\rangle\ Begin\_End\_Time, Intens\_Date; \\
& NVehicle, KReduct\ \ast\rangle\ Reduct\_Intens; \\
& Edge\_Prop_1,\ ...,\ Edge\_Prop_n\ \ast\rangle\ Edge; \\
& Edge,\ Reduct\_Intens, Intens\_Tram,\ Intens\_Ped, \\
& Begin\_End\_Time,\ Intens\_Date \ast\rangle\ INTENSITY
\end{aligned}
\tag{7}
$$

The «Density» traffic flow class is specified by the following basic parameters:

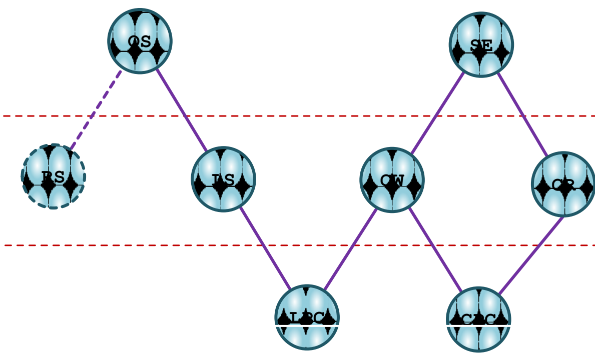

Fig. 7. Taxonomic tree of the class «Intensity»

start and end time of the density measurement (Begin_End_Time); range of values – from 00 hours 00 min. till 23 hours. 59 min.;

- date of measurement $Date\_On \in Date[YY:MM:DD]$; presentation format – YY: MM: DD;
- start time Begin_End_Time= $Time\_Begin \in Time\,[HH:MM:SS]$ and end $Time\_End \in Time\,[HH:MM:SS]$ of measurement. The range of values – from 00 hours 00 min. till 23 hours 59 min.

The combination of these factors determines the immanent properties of the class «Density»:

$$
\begin{aligned}
\forall density \,|\; & density\ \ is\_a\ \ DENSITY\ con\_of \\
& (Uch,\ Reduct\_Intens, \\
& Begin\_End\_Time,\ Date)\,| \\
& (uch\ \ is\_a\ \ Uch), \\
& reduct\_intens\ \ is\_a\ \ Reduct\_Intens\ , \\
& begin\_end\_time\ \ is\_a\ \ Time\ , \\
& (date\ \ is\_a\ Time)
\end{aligned}
$$

Taxonomic model of the class «Density» is as follows:

$$
\begin{aligned}
& Time\ \ast\rangle\ Begin\_End\_Time, Date; \\
& NVehicle, KReduct\ \ast\rangle\ Reduct\_Transp; \\
& Uch\_Prop_1,\ ...,\ Uch\_Prop_n\ \ast\rangle\ Uch; \\
& Uch,\ Reduct\_Transp, \\
& Begin\_End\_Time,\ Date \ast\rangle\ DENSITY
\end{aligned}
\tag{8}
$$

The model of the class «*Density*» taking into account these constructions will be determined by the taxonomic tree (Fig. 8).

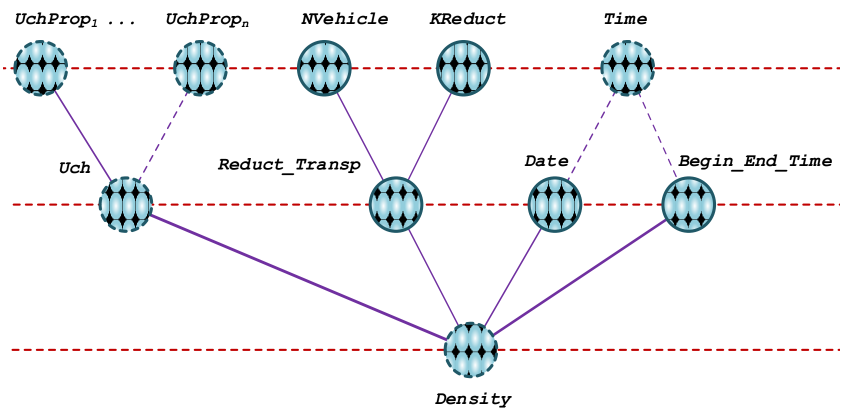

Fig. 8. Taxonomic tree of the class «Density»

VI. Conclusion

Basic mathematical models on the basis of the fundamental laws of intellectual control were analyzed. Decomposition of the underlying model is based on the information, methodological and functional principles of stratification. The strata of the problem-oriented mathematical models are set up in the form of sets of private models and sequences of its mappings – morphisms. The morphisms meet the system quality criteria and are suitable for transport management.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[2] Mikheev, S.V. Transport Infrastructure Management / T.I. Miheeva, S.V. Miheev, O.N. Saprykin. – Samara: Inteltrans, 2015. – 173 p.

[3] Cascetta, E. Transportation systems analysis : models and applications // NY : Springer, 2009. – 752 p.

[4] Patterns of designing complex systems / T.I. Mikheeva, S.V. Mikheev, O.K. Go¬lovnin, O.N. Saprykin. – Samara : Inteltrans, 2015. – 216 p.

[5] Inose H., Hamada T. Traffic management / M. : Transport, 1983. 248 p.

[6] Intelligent transport geographic information system ITSGIS. Core / T.I. Mikheeva, S.V. Mikheev, O.K. Golovnin, A.V. Sidorov, E.A. Savinov. – Samara : Inteltrans, 2016.–T.1.–171 p.

[7] Araghi, S. Optimal design of traffic signal controller, using neural networks and fuzzy logic systems / S. Araghi, A. Khosravi, D. Creighton // Proceedings of the International Joint Conference on Neural Networks 2014 (IJCNN) 6-11 July, 2014, Beijing, China. Pp. 42 47.

[8] Barcelo, J. Microscopic traffic simulation : a tool for the design, analysis and evaluation of intelligent transport systems / J. Barcelo [et al.] // Journal of Intelligent and Robotic Systems. – 2005. – T. 41, № 2 – 3. – P. 173 – 203.

[9] Benediktsson, J.A. Classification of hyperspectral data from urban areas based on extended morphological profiles / J.A. Benediktsson, J.A. Palmason, J.R. Sveinsson // IEEE Transactions on Geoscience and Remote Sensing. – 2005. – Vol. 43, No. 3. – P. 480 – 491.

[10] Mikheev, S.V. Razrabotka CAD traffic management systems / S.V. Miheev, T.I. Miheeva / Intelligent Systems: Proceedings of the Sixth International Symposium (INTELS'2004) / Edited by K.A. Pupkova. – M.: RUSAKI, 2004. P. 488–491.

[11] Bonzani, I. From experiments to hydrodynamics traffic flow models / I. Bonzani, L. Mussone // Math. Comput. Modeling. – 2003. – Vol. 37; T. 12 – 13. – P. 1435 – 1442.

[12] Cascetta, E. Transportation systems analysis : models and applications // NY : Springer, 2009. – 752 p.

[13] Mikheev, S.V. Automated traffic control system / S.V. Miheev, T.I. Miheeva, A.V. Zolotovickij // Maths. Computer. Education. Proceedings of the Institute of International Conf. – Pushchino – Moskva: MGU Progress – Tradiciya, – 2000. – P. 207–214.

[14] Mikheev, S.V. Design of traffic management systems based on the relations of inheritance of properties // Proceedings of the 6th international scientific-practical conference "Organization and safety of traffic in large cities" – SPb: SPbADI, – 2004. P. 241–245.

[15] Cavalli, R. Hyperspectral Sensor Data Capability for Retrieving Complex Urban Land Cover in Comparison with Multispectral Data : Venice City Case Study (Italy) / R. Cavalli, L. Fusilli, S. Pascucci, S. Piguatti, F. Santini // Sensors. – 2008. – Vol. 8. – P. 3299 – 3320