VOL. 10, NO 22, DECEMBER, 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

METAMODEL FOR SOFTWARE SOLUTIONS IN COMPUTED TOMOGRAPHY

Vitaliy Mezhuyev
Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Gambang, Malaysia
E-Mail: mejuev@ukr.net

ABSTRACT

Metamodelling approach is now widely used to automate development of software for the general purposes. This paper expands a "classical" metamodeling approach to the design of software for mathematical modelling. Paper defines the metamodel to solve different problems in computed tomography. The proposed approach was applied for the reconstruction of the structure of three-dimensional objects on a system of their traces on mutually perpendicular planes. It was also used for generation of software, intended for the search of illegal items during customs control.

Keywords: metamodel, domain-specific mathematical modelling, computed tomography, spline interpolation.

INTRODUCTION

In [1] the methodology for Domain-Specific Mathematical Modelling (DSMM) is proposed. It enhances Domain-Specific Modelling [2], used for the model driven software systems development.

DSMM allows us development of metamodels in the different mathematical semantics, which becomes possible due to introduction of an additional level of the metamodelling architecture [3]. DSMM tools [4] allow their users to define mathematical metatypes and apply them for the development of domain specific types.

One of the prominent examples is a geometrical meta-metamodel, which defines in the terms of sets the basic (corresponding to the dimensions of the space) geometrical metatypes (point, line, surface and 3D region). Geometrical meta-metamodel was used for the design of cyber-physical systems [5] and mathematical modelling surfaces of celestial bodies on the base of radiolocation data [6]. Its application for the design of software tools for physical modelling and simulation was considered in [7].

Formally, we define metamodel as a triple, which include an alphabet of types, a grammar, used to combine instances of the types, and applicable operations. Geometrical meta-metamodel includes alphabet of the metatypes, being the common result of abstraction from the geometrical structure of the physical objects. In this paper on the base of geometrical meta-metamodel we define Metamodel for Tomography (MT). Alphabet of MT includes the type of geometrical plane, created on the base of the metatype of a geometrical surface. To produce domain specific types, instances of the plane are attributed by specific data (functions of distribution of physical properties) obtained in the process of tomographic scan.

Operations of the geometrical meta-metamodel are built on the base of the methods of interpolation, interlination and interflatation [8]. In this paper, to restore a structure of three-dimensional objects we propose new interpolation operators, build on the base on interflatation of functions. Paper considers a practical implementation of the approach - computer tool Brain Reconstruction, which builds a three-dimensional image of brains by the system of their parallel sections. Conclusion and references list finalize the paper.

MATERIAL AND METHODS

In tomographic research there is often a problem arose by the set of given sections to obtain an image of a body in its any intersection [9]. This problem, for example, occurs in medicine and biology, at the study of the human or experimental animal's cerebral cortex.

Typically, the data, containing properties of a body, are obtained in the form of photographic images of the sections that are carriers of the flat distribution of intensity of the light wave. From it follows the appropriateness of the geometrical plane as a basic type to produce instances of computer objects as carriers of distribution of some physical property (density, light intensity, etc.).

The type of the plane in the metamodel MT is based on the metatype of geometric surface of the meta-metamodel G [5]. In terms of analytical geometry, we define the type of the plane by setting restrictions on the equation of the surface:

$$\Pi(x, y, z) = 0 \tag{1}$$

The equation of a plane surface is of the first order. In Cartesian coordinates

$$Ax + By + Cz + D = 0 \tag{2}$$

where A, B, C, D are some constants.

To define the type of the plane we put restrictions on the possible values of x, y, z. It is sufficient to set restrictions on two from the three variables, e.g.

$$\begin{cases} x_1 \le x \le x_2 \\ y_1 \le y \le y_2 \end{cases} \tag{3}$$

Here the mathematical part of the type definition is done. Next stage is to develop a domain specific type. The idea is to combine the mathematical type with domain specific properties, e.g. a function of a distribution of physical values. Such the distributions typically come from a tomograph as images in raster format, e.g. bitmap. Let us use this format due to the mathematical simplicity of the definition of function of distribution of physical data. To do it, each point of the plane with coordinates (x, y) we will consider as a parameter of the function, representing colour in some model (RGB, CMYB, HSB).

Let us develop the method to study distribution of a physical property $u(x, y, z)$ of an internal structure of a three-dimensional body (e.g., density) in any section. Without loss of generality we assume the body is fully located in the unit cube $D = [0,1]^3$.

The source of information about the function $u(x, y, z)$, i.e. the internal structure of the body, is its traces on the system of mutually perpendicular planes:

$$\begin{cases} x = \dfrac{i}{M_1}, i = \overline{0, M_1} \\ y = \dfrac{j}{M_2}, j = \overline{0, M_2} \\ z = \dfrac{k}{M_3}, k = \overline{0, M_3} \end{cases} \tag{4}$$

i.e. the set of functions

$$\begin{aligned} &u\left(\frac{i}{M_1}, y, z\right), i = \overline{0, M_1}; \\ &u\left(x, \frac{j}{M_2}, z\right), j = \overline{0, M_2}; \\ &u\left(x, y, \frac{k}{M_3}\right), k = \overline{0, M_3}. \end{aligned} \tag{5}$$

To restore the internal structure of a body we develop operators of spline interpolation O_1, O_2, O_3 of the function u by the variables x, y, z, correspondingly

$$\begin{aligned} O_1 u(x,y,z) &= \sum_{i=0}^{M_1} Sp_{M_1,i}(x) u\left(\frac{i}{M_1}, y, z\right), \\ O_2 u(x,y,z) &= \sum_{j=0}^{M_2} Sp_{M_2,j}(y) u\left(x, \frac{j}{M_2}, z\right), \\ O_3 u(x,y,z) &= \sum_{k=0}^{M_3} Sp_{M_3,k}(z) u\left(x, y, \frac{k}{M_3}\right), \end{aligned} \tag{6}$$

where $Sp_{M_1,i}(x)$ are the basic splines of degree $m\ (m = 1,2,3)$ having the properties

$$Sp_{M_1,i}\left(\frac{p}{M_1}\right) = \delta_{i,p},\ i, p = \overline{0, M_1} \tag{7}$$

Similarly we define splines $Sp_{M_2,j}$, $Sp_{M_3,k}$.

Interlination (interflatation) of a function of many variables is recovery (possibly, approximate) of this function by the help of its traces and traces of its derivatives up to the given order on the system of lines (or surfaces, respectively) [8].

The operator of spline-interflatation is defined as

$$Lu(x,y,z) = \begin{pmatrix} O_1 + O_2 + O_3 - O_1O_2 - \\ O_1O_2 - O_2O_3 + O_1O_2O_3 \end{pmatrix} u(x,y,z) \tag{8}$$

It has the following properties:

$$\begin{aligned} &Lu\left(\frac{i}{M_1}, y, z\right) = u\left(\frac{i}{M_1}, y, z\right), i = \overline{0, M_1}; \\ &Lu\left(x, \frac{j}{M_2}, z\right) = u\left(x, \frac{j}{M_2}, z\right), j = \overline{0, M_2}; \\ &Lu\left(x, y, \frac{k}{M_3}\right) = u\left(x, y, \frac{k}{M_3}\right), k = \overline{0, M_3}. \end{aligned} \tag{9}$$

In the real case, if experimental data are given with some accuracy, the following theorem can be formulated.

Theorem. If the experimental data

$$\tilde{u}\left(\frac{i}{M_1}, y, z\right), i = \overline{0, M_1};\ \tilde{u}\left(x, \frac{j}{M_2}, z\right), j = \overline{0, M_2}; \tilde{u}\left(x, y, \frac{k}{M_3}\right), k = \overline{0, M_3}$$

Are given with maximum error ε, and ψ is a random variable uniformly distributed in D, i.e.

$$\tilde{u}_{1,i}(y,z) = u\left(\frac{i}{M_1}, y, z\right) + \varepsilon\psi, i = \overline{0, M_1};$$

$$\tilde{u}_{2,j}(x,z) = u\left(x, \frac{j}{M_2}, z\right) + \varepsilon\psi, j = \overline{0, M_2};$$

$$\tilde{u}_{3,k}(x,y) = u\left(x, y, \frac{k}{M_3}\right) + \varepsilon\psi, k = \overline{0, M_3}$$

the operator

$$L\tilde{u}(x,y,z) = \begin{pmatrix} O_1 + O_2 + O_3 - O_1O_2 - O_1O_2 \\ -O_2O_3 + O_1O_2O_3 \end{pmatrix} \tilde{u}(x,y,z)$$

will have error $\mathrm{Ru}(x,y,z) = u(x,y,z) - L\tilde{u}(x,y,z)$, satisfying the following relationship $\|\mathrm{Ru}\|_{C(D)} = O\left(M_1^{-m-1} M_2^{-m-2} M_3^{-m-3}\right) + O(\varepsilon)$.

Proof.

$$\mathrm{Ru}(x,y,z) = u(x,y,z) - L\tilde{u}(x,y,z) =$$
$$= u(x,y,z) - L(u(x,y,z) + \varepsilon\psi(x,y,z)) =$$
$$= u(x,y,z) - L(u(x,y,z)) - \varepsilon\ L\psi(x,y,z).$$

From it follows inequality $\|\mathrm{Ru}\|_{C(D)} \le \|u - Lu\|_{C(D)} + \varepsilon\ \|L\psi\|_{C(D)}$, giving the proof.

PRACTICAL IMPLEMENTATION

Practical implementation of MT was done in the software tool Brain Reconstruction. Note, that the software framework, considered in [4; 7] includes Brain Reconstruction as a partial case. DSMM tools allows their users to define arbitrary metamodel within a general meta-metamodel *G*, including that for solving problems of computed tomography.

Figure-1 illustrates the interface of Brain Reconstruction, where we use the method of spline interflatation to reconstruct the three-dimensional image of human brains by the system of its parallel sections. Brain Reconstruction uses one type of the metamodel – a plane - to specify the distribution of physical properties (here, intensity of colours). The distribution a user does by attributisation of the plane by the function $u(x,y,z)$. This attributed plane is used for subsequent instantiation of objects that are directly used to build the model, i.e. the three-dimensional image of the cerebral cortex.

At the level of the metamodel, the attributing is a method that assigns to each element of the set of geometrical points on the plane the value of the function $u(x,y,z)$.

The input for Brain Reconstruction is a set of raster images (files in bitmap format), depicting the sections of a cerebral cortex on a system of mutually perpendicular planes $x = x_i, y = y_j, z = z_k$, $i = \overline{1..M_1}, j = \overline{1..M_2}, k = \overline{1..M_3}$. Names of files indicates their belonging to one of three parallel sections (for example, «x1.bmp», «x2.bmp», «x3.bmp» are the names of images, representing brain sections in a plane, perpendicular to the axis OX).

Brain Reconstruction reads the files that meet the following criteria:

a) The file format corresponds to the BMP standard.

b) The name begins with Latin letters x, y or z, reflecting a section belonging to one of mutually perpendicular planes.

c) The next (after Latin letters) characters are Arabic numbers, which are used to determine the serial number of the Figure in the corresponding system of planes.

All correct files are recorded in the array of figures - instances of the geometrical type plane, and their number - in the relevant internal variables M_1, M_2, M_3.

Thus, the structure of the model is automatically built by analysing the input information (files of images). Under the structure of the model, we refer to the dimension of the space and the number of elements in the array of images. Based on this information the model of the space $D = [0,1]^3$ is defined and the distances between separate planes are calculated.

Our method allows a user to recover the three-dimensional structure of the body in arbitrary distances between its parallel sections. For simplification of explanations and to be close to the practice of scanning tomographic images (technically it is easier to obtain intersections with a constant step), let us assume that the distance between intersections is the same for any of the systems of planes.

Operators of spline-interflatation (6) are implemented as software functions *O1, O2, O3, O12, O13, O23, O123*, parameters of which are variables *x, y, z* of high accuracy (the long double type of C++). Functions *O12, O13, O23, O123* use the values that return operator functions *O1, O2, O3*. Functions *O1, O2, O3* compute the closest to the point with coordinates (x, y, z) layers *i, j, k* and using them refer to specific elements of the array of images.

VOL. 10, NO 22, DECEMBER, 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

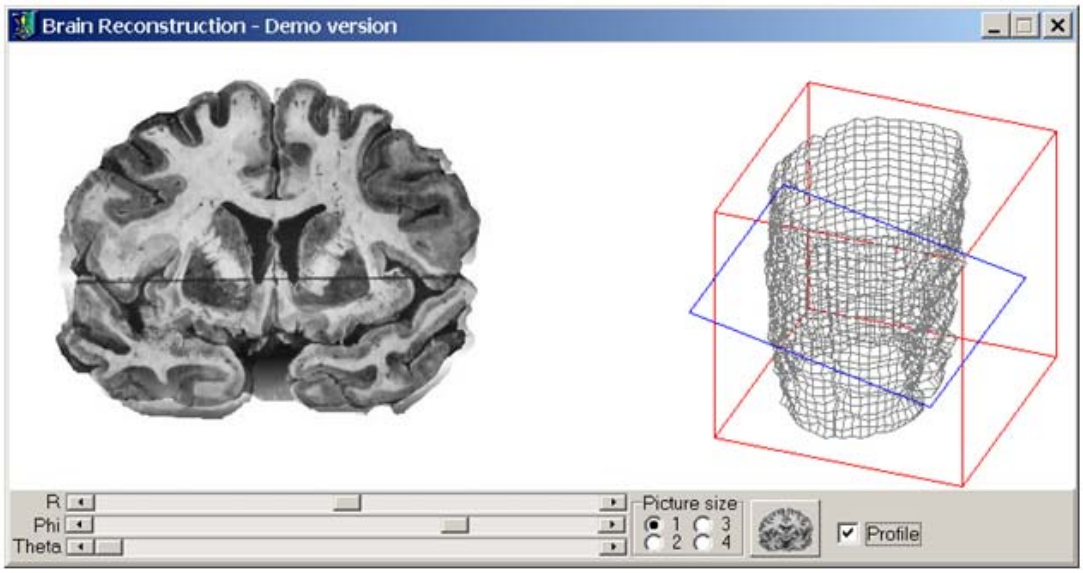

Figure-1. Interface of the program "Brain Reconstruction".

For example, if the calculated number of the layer is *i* and this is OZ intersection, then the reference to the element of the array *Images* is as follows:

Images[1][i]->Pixels[x][y]

Every image is a two-dimensional array of pixels (picture elements), storing an attribute - colour. We use RGB colour model, according to which any colour can be represented as a superposition of three basic colours - Red, Green and Blue.

Each component of the colour - Red, Green or Blue is given by one byte, that can reflect $2^8=256$ shades of the colour (the intensity varies from 0 to 255, the 255 reflects the largest intensity of the colour). The total number of colours that can be set by 3 bytes is 256^3=16 777 216, which is close to the colour recognition possibility of a human eye (so called True Colour model).

For further discussion let us use hexadecimal number system, which sign in C++ is a combination of characters *0x*. Fully set bytes (FF) show the highest intensity of the corresponding colours' component (Table-1).

Table-1. The value of bytes in the RGB colours' model.

Colour	1 byte	2 byte	3 byte	Value
Red	00	00	FF	0x0000FF
Green	00	FF	00	0x00FF00
Blue	FF	00	00	0xFF0000

Thus, 0xFFFFFF is the value with the highest intensity of all colours' components and so giving white, the value 0x000000 is black. The range between 0x000000 and 0xFFFFFF having equal values of each colour' component allows us operate with shades of the grey colour (e.g., 0x111111 or 0xAAAAAA).

Software implementation of operators *O1, O2, O3* allocate each of the RGB colours components separately. The functions *O1, O2, O3* perform a selection of individual colours' component at this point. E.g.

R = Images[1][i]->Pixels[x][y]&0x0000FF

performs selection of the red colour component of a pixel that belongs to the perpendicular to the axis OZ plane *i* and has coordinates (x, y).

To separate individual colours components we use bitwise *and* operation (&), which clears all other bits besides given by the second operand (in above sample it is *0x000000FF*). To explain this operation let us use binary system. For example, we have the following colour value in the binary form 10101010101010101010101010.

Taking that $000000FF_{16}$ = $000000000000000011111111_2$ (each byte is set by 8 binary digits, i.e. to set the colour as a combination of three components we need 24 bits). To set a single number in hexadecimal format we need 4 bits.

Using bitwise "and" (&) operation we allocate a separate component of a colour

10101010101010101010101010
<u>000000000000000011111111</u>
000000000000000**10101010**

In such a way, the selection of components of a colour is made (in this example the right byte, which corresponding to red colour).

ARPN Journal of Engineering and Applied Sciences

For the green colour component we may use

G= Images[1][i]->Pixels[x][y]&0x0000FF00

After calculation of all separate components, we can form the colour again

Color = R + G + B

METHODS OF 3D GRAPHICS

The metamodel MT also includes methods of three-dimensional graphics to support visualisation and analyses. To build a three-dimensional perspective view the set of coordinates of the real object are transformed into the screen coordinates. This process is carried out in two stages:

a) Real (world) coordinates of each point (x, y, z) of an object are transformed into specific coordinates (x_v, y_v, z_v).
b) Specific coordinates (x_v, y_v, z_v) of each point of an object using a perspective transformation are converted into screen coordinates (x_e, y_e).

To support a user in solution of this problem, the metamodel MT gives the software functions (API), defining relationship between sets of the real and specific coordinates. Geometrically, to build an image of a three-dimensional object, we need make its projection on the plane of a screen. For this purpose, we introduce a specific coordinate system in which an object is projected on a plane $X_v0_vY_v$, and axis Z_e0_e determines the direction of observation (see Figure-2).

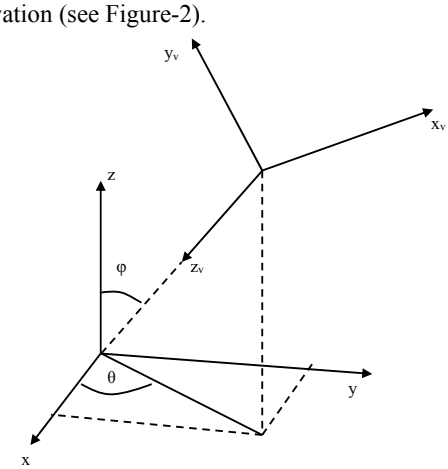

Figure-2. 3D to 2D transformation.

To perform the specific transformation the observation point also to be set. Metamodel defines a point of observation in the spherical coordinate system by following parameters:

θ = angle of the axis OX with projection of the observation line on the plane XOY;
φ = polar angle of the observation line with OZ axis;
ρ = distance to the observation point.

Setting the triple of parameters (θ, φ, ρ) allows us to make a 3D transformation of a geometrical object. Let the position of a point in three-dimensional space set by coordinates (x, y, z). Then the specific transformation we can write in the form:

$$|x_v\ y_v\ z_v\ 1| = |x\ y\ z\ 1|\cdot V \qquad (10)$$

where

$$V = \begin{vmatrix} -\sin\theta & -\cos\varphi\cos\theta & -\sin\varphi\cos\theta & 0 \\ \cos\theta & -\cos\varphi\sin\theta & -\sin\varphi\sin\theta & 0 \\ 0 & \sin\varphi & -\cos\varphi & 0 \\ 0 & 0 & \rho & 1 \end{vmatrix} \qquad (11).$$

Coordinates x_v and y_v of a point of an object we can already use to build a screen image. In this case we will obtain the orthogonal projection, where each point P of an object is projected into the point P' by drawing the line, perpendicular to the plane XOY (i.e. parallel to the observation line).

Such type of projection has place at moving the observation point into infinity, where parallel lines of a three-dimensional object remain parallel at its projection as well.

OTHER APPLICATIONS OF THE METAMODEL

The metamodel MT was also used for the development of software tools for solving problems in other domains of computed tomography. Figure-3 illustrates this approach for search of unauthorized items during customs control.

The geometrical plane is here the basic type of the metamodel, used to build the data models in a software. The plane is a carrier of physical properties - obtained from a tomograph roentgen images of distribution of density of a car, which serves as an input for applications of a mathematical method (fixed in the patent [10]).

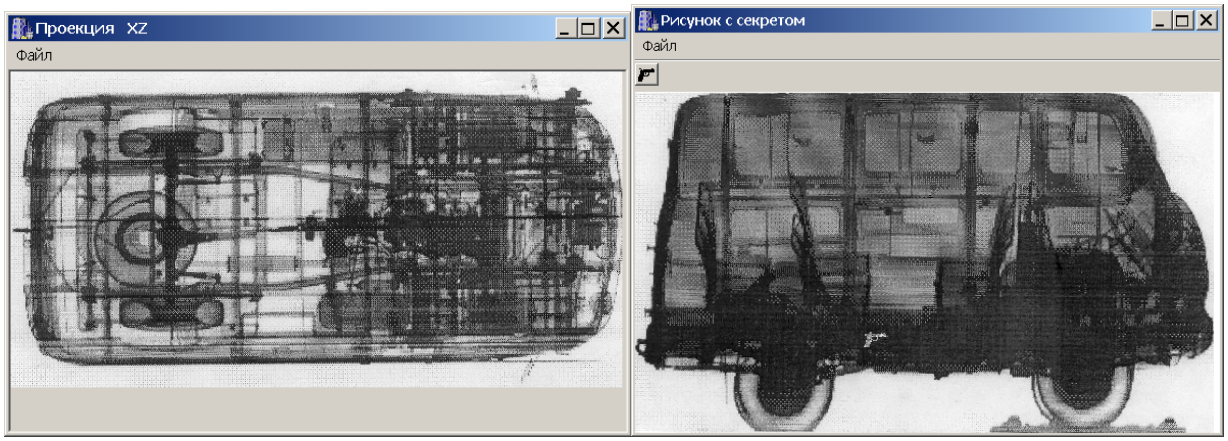

Figure-3. Use of metamodel to generate software for search of illegal items during customs control.

CONCLUSIONS

Based on the geometrical meta-metamodel the metamodel MT was defined, allowing design of software for solving problems of computed tomography. Alphabet of MT includes the type of geometrical plane, which inherits the metatype of the surface. Instances of the plane are attributed by functions of distribution of physical properties obtained in the process of tomographic scanning.

The approach was proven by development of a software tool that reconstructs a three-dimensional image of a brain by the system of its mutually perpendicular sections. The method restores the structure of three-dimensional objects using interpolation operators, based on interflatation of functions. The proposed metamodel MT was also used to generate software for the search of illegal items during customs control.

REFERENCES

[1] Vitaliy Mezhuyev. 2014. Methodology of Domain Specific Mathematical Modelling. Proceedings of the 3rd International Congress on Natural Sciences and Engineering ICNSE 2014 (Kyoto, Japan, May 7-9). pp. 54-64.

[2] Steven Kelly and Juha-Pekka Tolvanen. 2008. Domain-Specific Modeling: Enabling Full Code Generation. Wiley-IEEE Computer Society Pr. p. 444.

[3] Vitaliy Mezhuyev. 2015. Metamodelling Architecture for Modelling Domains with Different Mathematical Structure. Advanced Computer and Communication Engineering Technology. Lecture Notes in Electrical Engineering. 315: 1049-1055.

[4] Vitaliy Mezhuyev. 2014. Architecture of software tools for Domain-Specific Mathematical Modelling. Proceedings of 2014 International Conference on Computer, Communication, and Control Technology (Langkawi, Malaysia, September 2-4). pp. 166-170.

[5] Vitaliy Mezhuyev, Refik Samet. 2013. Geometrical Meta-metamodel for Cyber-Physical Modelling. Proceedings of International Conference Cyberworlds 2013 (Yokohama, Japan, October 21-23). pp. 89-93.

[6] Vitaliy Mezhuyev, Oleg M. Lytvyn, Jasni Mohamad Zain. 2015. Metamodel for Mathematical Modelling Surfaces of Celestial Bodies on the Base of Radiolocation Data. Indian Journal of Science and Technology. 8(13): 70630. pp. 1-8.

[7] Vitaliy Mezhuyev, Felipe Pérez-Rodríguez. 2015. Metamodelling approach and software tools for physical modelling and simulation. International Journal of Software Engineering and Computer Sciences (IJSECS). 1: pp. 1-13.

[8] Oleg M. Lytvyn. 2002. Interlination of functions and some its applications. Kharkiv: Osnova. p. 544.

[9] Frank Natterer. 2001. The Mathematics of Computerized Tomography. Classics in Applied Mathematics. Book 32. SIAM: p. 222.

[10] Patent on invention No. 78568, Ukraine. The method to restore the internal structure of a 3D object. Ivan Sergienko, Oleg Lytvyn, Vitaliy Mezhuyev. - №2 0041210438. - Decl. 20.12.2004. - Publ. 10.04. 2007. – Issue No. 4.