DE GRUYTER OPEN

Management Systems in

2016, No 3 (23), pp 145-149

Date of submission of the article to the Editor: 02/2016
Date of acceptance of the article by the Editor: 04/2016

DOI 10.12914/MSPE-01-03-2016

THE PROPOSAL OF A EVOLUTIONARY STRATEGY GENERATING THE DATA STRUCTURES BASED ON A HORIZONTAL TREE FOR THE TESTS

Marek ŻUKOWICZ, Michał MARKIEWICZ
Rzeszow University of Technology

Abstract:
The aim of the article is to present a mathematical definition of the object model, that is known in computer science as TreeList and to show application of this model for design evolutionary algorithm, that purpose is to generate structures based on this object. The first chapter introduces the reader to the problem of presenting data using the TreeList object. The second chapter describes the problem of testing data structures based on TreeList. The third one shows a mathematical model of the object TreeList and the parameters, used in determining the utility of structures created through this model and in evolutionary strategy, that generates these structures for testing purposes. The last chapter provides a brief summary and plans for future research related to the algorithm presented in the article.

Key words: *TreeList object, evolutionary algorithms, evolutionary strategy, software testing*

INTRODUCTION

Very often in science, technology, and teaching it exists a situation, that an occurrence or a process which can be described by the model, which is close to a specific structure in the nature. The article focuses on the description of data storage based on structure called tree. In the next three chapters will be discussed the horizontal trees, that are classified as traditional trees. These chapters presents mathematical definition of the structures based on TreeList and essence of the problem raised in the publication in order to maintain consistency of the article.

Data trees are a class of graphs known in mathematics. From the point of view of software testing for usability and presentation the data, a very important aspect is the arrangement of these data and an overall presentation of data structures. For the user convenient and transparent way of data presentation are structures called TreeList [7, 8]. Properties of the TreeList structure and a classic tree known in informatics [1] are analogical, but with one difference. TreeList structures are used for the horizontal display of pieces of data, that are dependent on each other in a hierarchical manner. Examples of such tree presents a Figure 1.

Numer	Nazwa pozycji	Wartość pozycji	Wartość przychodów
1	**ROBOCIZNA**	**301 199,25 zł**	**436 738,91 zł**
1.1	**ROBOCIZNA WŁASNA**	**299 199,25 zł**	**433 838,91 zł**
1.1.1	Płace	45 645,00 zł	66 185,25 zł
1.1.2	Narzuty płacowe	45 654,00 zł	66 198,30 zł
1.1.3	Narzędzia osobiste	2 343,00 zł	3 397,35 zł
1.1.4	Przewozy pracowników	2 796,00 zł	4 054,20 zł
1.1.5	Delegacje - noclegi	13 424,50 zł	19 465,52 zł
1.1.6	Delegacje - diety i ryczałty	34 670,00 zł	50 271,50 zł
1.1.7	Posiłki i napoje regeneracyjne	2 249,00 zł	3 261,05 zł
1.1.8	Koszty BHP, odzieży	67 500,00 zł	97 875,00 zł
1.1.9	Badania okresowe	2 553,60 zł	3 702,72 zł
1.1.10	Szkolenia pracowników	76 700,00 zł	111 215,00 zł
1.1.11	Ekwiwalent za pranie	5 664,15 zł	8 213,02 zł
1.2	**ROBOCIZNA OBCA**	**2 000,00 zł**	**2 900,00 zł**
1.2.1	Umowy zlecenia - płace	1 000,00 zł	1 450,00 zł
1.2.2	Umowy zlecenia - narzuty płacowe	1 000,00 zł	1 450,00 zł
2	**MATERIAŁY**	**0,00 zł**	**0,00 zł**
3	**SPRZĘT, TRANSPORT**	**3 000,00 zł**	**4 350,00 zł**
3.1	Najem sprzętu	1 000,00 zł	1 450,00 zł
3.2	Usługi transportowe	1 000,00 zł	1 450,00 zł
3.3	Zakup sprzetu, amortyzacja	1 000,00 zł	1 450,00 zł
4	**PODWYKONAWCY**	**4 000,00 zł**	**5 800,00 zł**
5	**KOSZTY ORGANIZACJI BUDOWY**	**46 000,00 zł**	**66 700,00 zł**

Fig. 1 Tree showing costs in the B2B OPTIbud system authorship of OPTeam

The kind of trees as in Figure 1 can be obtained simply by placing trees described in [1] from the left side to the right side of the screen. This gives a good visual effect of a horizontal presentation. The structure presented above is usually implemented in this way, that the user can expand and collapse branches (nodes). This is useful in the data presentation that are related with costs, because the costs do not always have to be showed at the analytical level. The above tree is an example of a presentation of costs. The data, that contains the tree can be arbitrary, but it is important, that the data has to be hierarchically dependent.

TESTING OF THE HORIZONTAL TREES PROBLEM

A very important step in designing tests is to prepare test data (input data) and structures for testing. The article focuses mainly on the generation of structures designed to receive and to process multidimensional input data. Such procedure is an activity, that has to be completed before delivery of the finished input data to the tree structure. These structures should be best suited to the specifications of the tested program. The article presents the approach, that using an evolutionary algorithm and given initial structures to creates new better structures suited to the needs of the structure test, basing on the example of using class TreeListControl in designing construction budgets in the B2B OPTIbud system. Evolutionary algorithms have a lot of features that allow in some way to control evolutionary process [6] and thus there is a high probability that generated structures will be characterized by such features as [3]:

- dependence on the system (quality depends on technological aspects),
- completeness (the degree to which this data has value for all attributes in the system),
- timeliness,
- credibility,
- cohesion (eg. a consistency between the test data),
- accuracy (for tests on numbers in applications that require high accuracy, eg. in the conversion of units of measurement),
- dependence on context.

All features have to be satisfy for the test data [2, 4, 5]. There are not many such publications that would describe the strategy of testing data structures based on horizontal trees. Therefore, the authors decided to describe this problem and suggest a solution, that allows to generate horizontal trees in this way that as the result of evolutionary processes there are created the structure corresponding to the preferences of people designing tests.

MATHEMATICAL DESCRIPTION OF STRUCTURES BASED ON THE HORIZONTAL TREES AND MEASURES OF THEIR USABILITY

Durning a designing the evolutionary algorithm, it must to be first make an assumptions about the data structure based on the horizontal tree and about parameters which can be measured in order to formulate a fitness function.

The assumptions concerning of the structure D consisting of horizontal trees are as follows:

- there is at least one tree (presenting the horisontal data),
- a node (including the root) has at least one feature which is described by a number (real positive number),
- the value of the specific node numer features in the level is a sum of the children of this node, when the children have only the (n + 1) level,
- any number feature of the node x (root too) can be divided (but it is not required) into periods with values x_i, and in this case the sum values for each period is equal, namely:

$$x = \sum_{i=1}^{m} x_i \qquad (1)$$

The possibilities of generating the above-described structures, which arise from assumptions are infinite. In practice in the information systems every structures are finite. Therefore it is needed a strategy that will provide the structure, which likely will be applied in practice. To achieve this goal, it should be specify measures related to the above-described structure. Before a generating structures will be taken into account the following restrictions:

- quantity of trees k in the structure D,
- a nesting level s_k of k-th tree (the level on which the leaves of a tree),
- the total number of children t_{kj} for a node j in k-th tree,
- number of numerical characteristics u – common ownership for all trees in the D,
- number of nonumerical characteristics v – common ownership for all trees in the D too.

Encoding information describing a single tree will be done by the vector:

$$d_k = \left[s_k \{t_{k1},\ldots, t_{ksk}\}\right] \qquad (2)$$

where:
d_k – number of a tree,
s_k – the nesting level of k-th tree,
t_k – the total number of children t_{kj} for a node j in k-th tree, (root,s level equals 0).

Let a single tree will be defined by the following function F_k measuring usability:

$$F_k(d_k) = \frac{1}{w_1 + w_2}\left[w_1 f(s_k) + \frac{w_2}{s_k}\left(\sum_{j=1}^{s_k} \frac{g_j(t_{kj})}{2}\right)\right] \qquad (3)$$

where:
functions f, g : N → (0,1] and w_1, w_2 are weights.

It is easy to see, that the funktion F_k also has a value in the range (0,1]. The structure, which will consist of the sum of individual trees will have the following description:

$$D_i(u_i, v_i) = \left\{\begin{array}{l} [\ s_1, \{t_{11}, \ldots t_{1s_1}\}] \\ [s_2, \{t_{21}, \ldots t_{2s_2}\}] \\ [s_k, \{t_{k1}, \ldots t_{ks_k}\}] \end{array}\right\} \qquad (4)$$

where:
i – numer of the structure consisting of horizontal trees,
u_i, v_i – numerical and nonnumerical characteristics common for all trees in the structure D_i.

This structure will be called later in the article the matrix of a structure of horizontal trees.

Let D = {D_1, … ,D_n} will be a set of a structures which constist of a horizontal trees.

The fitness function F : D → (0,1] for a structure D_i will be given by the formula:

$$F[D_i(u_1, v_1)] = \frac{w_d\left(h(u_i) + l(v_i)\right) + \frac{w_d}{n}\left(\sum_{k=1}^{n} F_k(d_k)\right)}{w_c + w_d} \qquad (5)$$

where:
w_c, w_d – weigths of a function F,
n – the numer of a trees in a structure $D_i(u_i\ v_i)$.

Having defined the fitness functions it can be designed an evolutionary algorithm, which will be described in the next chapter [6].

THE EWOLUTIONARY ALGORITHM PROJECT FOR D SET

In evolutionary algorithms, there is the phenomenon of crossing components and their mutation [6]. Therefore, to design an evolutionary algorithm there are needed two functions: one is responsible for crossing elements to one another and the second is the mutation operator. Therefore, according to the algorithm: KM Algorithm (crossing and mutation of horizontal trees). Let D_1, D_2 will be structures consist of horizontal trees. Let D_1 will be consist of trees and D_2 will be consist of trees. An algorithm KM is given as a function

$$X(D_1, D_2): D \times D \to D \tag{6}$$

by the conditions:

1. Take the rest of the division of n by 2, add to the rest 1, save the calculated value as x.
2. Take the rest of the division of k by 2, add to the rest, save the calculated value as y.
3. Create a new structure by combining x initial trees, starting from the first row in the matrix of a structure D_1, with rows of the structure D_2 starting from the row y.
4. Write down the number of trees in the new structure as s.
5. For each row in the newly established structure write numbers $(t_{i1}),\ldots,(t_{ik})$ binary.
6. For (i=1; i≤s; s++)
7. do {
8. k = 1;
9. Do a mutation of element t_{ik} with probability q, bit by bit.
10. If the mutated element t_{ik} is located in the field of the corresponding function, accept a mutation.
11. If the mutated element t_{ik} is not located in the field corresponding function, dont't accept a mutation.
12. k ++.
13. } while (t_{ik} exists in the tree s).
14. The guantity of numerical characteristics from structures D_1, D_2, choose randomly from values u_1, u_2, and save as u.
15. The guantity of nonnumerical characteristics from structures D_1, D_2, choose randomly from values v_1, v_2, and save as v.
16. Save newlu structure with parameters u and v.

Having defined an algorithm for implementing the process of crossover and mutation, it can be to specify evolutionary algorithm, in which the field is set D: ALGEN Algorithm (w_1, w_2, w_c, w_d, q, n, m, c, fv) (Evolutionary algotithm for structures D_1, … ,D_n)

{

1. Draw m structures and mark them as D'_1, D'_2, D'_3, … ,D'_m
   for (j=1, j < c, j + +)
   {
2. Do newly structures by using the KM Algorithm:
   $D''_1=X(D'_1, D'_2)$,
   $D''_2=X(D'_2, D'_3)$,
   $D''_3=X(D'_3, D'_4)$,
   .....
   $D''_m=X(D'_m, D'_1)$,
3. Compare the sum $s=\sum_{i=1}^{m} F[D'_i(u_i, v_i)]$ with the sum $l=\sum_{i=1}^{m} F[D''_i(u_i, v_i)]$
   – if s ≥ l, start next iteration,
   – if s < l change elements D'_1, D'_2, D'_3, … ,D'_m to D''_1, D''_2, D''_3, … ,D''_m,
   – if l > fv break iteration and save elements D'_1, D'_2, D'_3, … ,D'_m
   }
4. Give generated ofspring D'_1, D'_2, D'_3, … ,D'_m.

}

where:
w_1, w_2, w_c, w_d – weights of a function measuring usability (described in the previous section) ,
q – mutation probability,
n – numer of all structures,
c – numer of iterations in the case of failure,
fv – values, which break algorithm with positve result. This parameter determines whether to generate structures in the next iteration or not. It should also be noted that the values l, v can not be greather than number m.

APPLICATION AN ALGEN ALGORITHM IN THE BUDGET TESTING IN THE B2B OPTIBUD SYSTEM

Budgets in the B2B OPTIbud system

The Budgets module in the B2B OPTIbud system is used to create estimates and has been created for Polish construction companies. Its basic functionality: create budgets for projects on the basis of data from multiple sources, converting to budgets and schedules reflected on the timeline of the project [3]. Built-in module will be grouping mechanisms, merge, breaking the budget items and the ability to create all kinds of budgets: simplified or full, tracking the degree of implementation of the project the cost side as well. The main form of the budget simplified B2B system OPTIbud shown in the Figure 2.

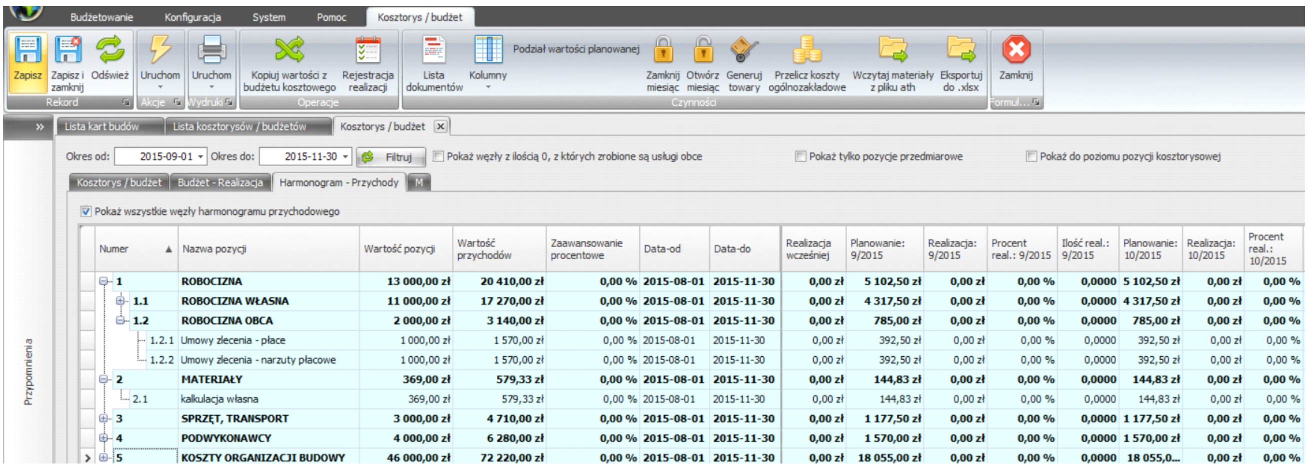

Fig. 2 Costs form of a building budget in the B2B OPTIbud system

A structure which presetns costs is analogous to the structure described in section mathematical description of structures based on the horizontal trees and mesures of their usability. This structure is a good example of an application of the ALGEN algotithm. Numerical characteristics shown in the picture are columns: Value of incomes (Wartość pozycji), Value of income (Wartość przychodów), Percentage advancement (Zaawansowanie procentowe), Earlier realization (Realizacja wcześniej), Realization 9/2015 (Realizacja 9/2015), Planning 9/2015 (Planowanie 9/2015), Percentage of realization 9/2015 (Procent real.: 9/2015),itd. Nonnumerical characteristics shown in the picture are columns Number (Numer), Name of a position (Nazwa Pozycji), Date from (Data-od), Date to (Data-do). The columns beginning with the words Planowanie are values created by dividing the planned revenues from the budget by 4, because the budget has that schedule (Date form and Date to). Quantity of a the trees equals 5 in this budget. Level of a tree, which have root with numer 1 equals 2 (root – level 0, nodes 1.1 and 1.2 – level I, so nodes 1.2.1 i 1.2.2 – level II). In the previous chapter are given definitions of a functionswhich measure the fitness of a tested structures. Formulas (3) and (5) contain functions f, g, h, l. The following there are given definitions of these functions, which will be used to implement KM and ALGEN algorithms on the system OPTIbud:

1. $$f(x)=-\frac{1}{4}|x-4|+1, \quad \text{for} \quad 1 \le x \le 8 \tag{7}$$

2. $$g_j(x)=\begin{cases} -\frac{1}{3}|x-3|+1, & \text{for } 1 \le x \le 6, j=1 \\ -\frac{1}{6}|x-6|+1, & \text{for } 3 \le x \le 12, j=2 \\ -\frac{1}{12}|x-12|+1, & \text{for } 8 \le x \le 24, j=3 \\ -\frac{1}{24}|x-24|+1, & \text{for } 12 \le x \le 48, j=4 \\ -\frac{1}{30}|x-30|+1, & \text{for } 20 \le x \le 8, j>4 \end{cases} \tag{8}$$

$$\begin{aligned} h(u)&-\frac{1}{6}|u-6|+1, \quad \text{for} \quad 1 \le u \le 12 \\ l(v)&-\frac{1}{3}|v-3|+1, \quad \text{for} \quad 1 \le v \le 6 \end{aligned} \tag{9}$$

The formulas of the above-described functions have been deliberately chosen in such a way that did not reach values greater than 1 and an evolutionary algorithm generates the best possible structure for testing (do not have a huge number of nodes).

An implementation of an ALGEN algorithm in the Budgets module in the system B2B OPTIbud

Let the initial input population D will consist of the following budgets:

$$D_1=\begin{Bmatrix}[4,\{1,2,4,8\}]\\ [3,\{3,7,18\}]\\ [4,\{5,6,12,24\}]\\ [2,\{7,8\}]\end{Bmatrix}, \quad D_2=\begin{Bmatrix}[4,\{1,2,4,8\}]\\ [3,\{3,7,18\}]\\ [4,\{5,6,12,24\}]\\ [2,\{7,8\}]\end{Bmatrix},$$

$$D_3=\begin{Bmatrix}[3,\{2,12,24\}]\\ [2,\{6,8,16\}]\\ [3,\{4,8,12\}]\\ [4,\{2,4,8,16\}]\end{Bmatrix}, \quad D_4=\begin{Bmatrix}[2,\{9,92\}]\\ [2,\{7,29\}]\\ [2,\{4,60\}]\\ [2,\{2,18\}]\end{Bmatrix},$$

$$D_5=\begin{Bmatrix}[2,\{1,4\}]\\ [2,\{2,3\}]\\ [3,\{4,20,60\}]\\ [3,\{2,9,16\}]\end{Bmatrix}, \quad D_6=\begin{Bmatrix}[3,\{9,27,92\}]\\ [3,\{7,16,29\}]\\ [2,\{4,60\}]\\ [2,\{2,18\}]\end{Bmatrix},$$

$$D_7=\begin{Bmatrix}[4,\{1,2,9,92\}]\\ [5,\{1,2,7,17,29\}]\\ [4,\{4,8,16,60\}]\\ [3,\{2,18,44\}]\end{Bmatrix}, \quad D_8=\begin{Bmatrix}[3,\{9,9,92\}]\\ [3,\{2,7,29\}]\\ [3,\{2,4,60\}]\\ [3,\{3,18,44\}]\end{Bmatrix},$$

$$D_9=\begin{Bmatrix}[2,\{9,92\}]\\ [2,\{7,33\}]\\ [2,\{4,44\}]\\ [2,\{2,18\}]\end{Bmatrix}, \quad D_{10}=\begin{Bmatrix}[4,\{1,5,9,92\}]\\ [3,\{7,16,29\}]\\ [2,\{4,60\}]\\ [3,\{2,7,18\}]\end{Bmatrix},$$

respectively with parametrs
u_1=6, v_1=2, u_2=8, v_2=2, u_3=7, v_3=3, u_4=7, v_4=3, u_5=4, v_5=2, u_6=5, v_6=2, u_7=6, v_7=2, u_8=7, v_8=3, u_9=8, v_9=3, u_{10}=10, v_{10}=4.
Let the input parameters of the ALGEN algorithm will have values:
w_1=2, w_2=3, w_c=2, w_d=1, q=0,2, n=10, m=4, c=30, fv=3.7.
The results returned by the algorithm after six iterations are:

$$D'_1=\begin{Bmatrix}[4,\{1,3,4,13\}]\\ [3,\{2,3,3\}]\\ [4,\{1,4,15,5\}]\\ [2,\{3,2\}]\end{Bmatrix}, \quad D'_2=\begin{Bmatrix}[2,\{5,46\}]\\ [3,\{2,3,3\}]\\ [3,\{1,4,4\}]\\ [4,\{2,2,1,2\}]\end{Bmatrix},$$

$$D'_3=\begin{Bmatrix}[4,\{1,2,6,14\}]\\ [5,\{1,1,4,17,20\}]\\ [4,\{4,3,2,8\}]\\ [2,\{3,5\}]\end{Bmatrix}, \quad D'_4=\begin{Bmatrix}[3,\{3,1,19\}]\\ [3,\{3,11,22\}]\\ [2,\{6,12\}]\\ [2,\{1,23\}]\end{Bmatrix},$$

with parameters u'_1=6, v'_1=3, u'_2=8, v'_2=2, u'_3=7, v'_3=3, u'_4=7, v'_4=3,
with values s=3.5003, l=3.7223.

For example, the first tree in the budget D'_1 in the B2B OPTIbud system is shown in the Figure 3.

There are an infinite number of possibilities for testing the ALGEN algorithm. The aim of the article, however, was to identify the methodological approach, which is why the further results related to the study of algorithm parameters will be released in the next publication.

CONCLUSIONS

The aim of this article is to describe the structures presenting the data and to show evolutionary algorithm, that generates these structures to obtain best fitness of them for tests. This was achieved by designing a mathematical model of the structure, that consists of a horizontal trees. Matching functions have been deliberately defined in this way, that their values are in the range (0,1] and can be easily converted into the percentages value.

In the future, the authors intend to extend research of the issue described in this work by modifying the algorithm ALGEN and compare its performance with the new one. The authors plan also to publish an article showing the behavior of the algorithm for different input data.

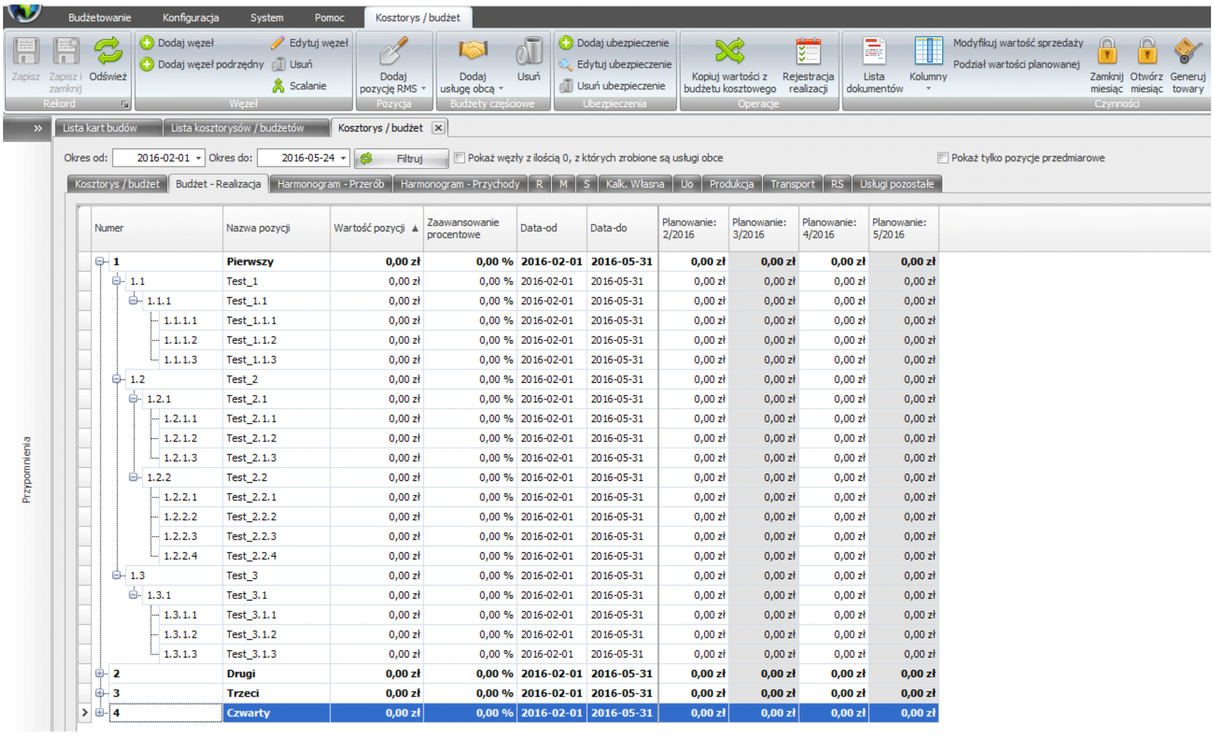

Fig. 3 Costs form in the budgets module D'$_1$ in the B2B OPTIbud system

REFERENCES

[1] L. Banachowski, K. Diks and W. Rytter. *Algorytmy i struktury danych*, Warszawa: Wydawnictwa Naukowo-Techniczne, 2011.

[2] D. Farley and J. Humble. *Ciągłe dostarczanie oprogramowania*, Gliwice: Helion, 2015.

[3] M. Łobaziewicz. „Standard architektury modelu systemu B2B wspomagającego zarządzanie procesami budowlanymi", in *Od procesów do oprogramowania: badania i praktyka*, P. Kosciuszenko, M. Śmiałek and J. Swacha, Warszawa: Wydawnictwo Polskie Towarzystwo Informatyczne, 2015, pp. 111-120.

[4] A. Piaskowy and R. Smilgin. *Dane Testowe: teoria i praktyka*, Gliwice: Helion, 2011.

[5] A. Roman. *Testowanie i jakość oprogramowania*, Warszawa: Wydawnictwo Naukowe PWN, 2015.

[6] D. Rutkowska, M. Piliński and L. Rutkowski. *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*, Warszawa: Wydawnictwo Naukowe PWN, 1997.

[7] WinForms Tree List, [Online]. Available: https://www.devexpress.com/products/net/controls/winforms/tree_list/

[8] WPF Tree List, [Online]. Available: https://www.devexpress.com/products/net/controls/wpf/tree_list/

mgr Marek Żukowicz, mgr Michał Markiewicz
Rzeszow University of Technology
Faculty of Electrical Engineering and Computer Science
Department of Automation and Computer Science
ul. Wincentego Pola 2, 35-959 Rzeszów, POLAND
e-mail: bobmarek@o2.pl
mmarkiewicz13@gmail.com

Artykuł w polskiej wersji językowej dostępny na stronie internetowej czasopisma.

The article in Polish language version available on the website of the journal