On Multi-Objective Optimization Aided Visualization of Graphs Related to Business Process Diagrams

Vytautas JANČAUSKAS [a], Giedrius KAUKAS [b], Antanas ŽILINSKAS [a,1] and Julius ŽILINSKAS [a]

[a] *Institute of Mathematics and Informatics, Vilnius University, Lithuania*

[b] *ORGSOFT, Vilnius, Lithuania*

Abstract. A problem of the drawing of aesthetically looking graphs, related to business process diagrams, is considered. We model a situation where sites of flow objects of the diagram are fixed, and the sequence flow is defined. The edges of a graph, which represent the sequence flow, should be drawn aiming at an aesthetical image. The latter problem is reformulated as a multi-objective combinatorial optimization problem. The generally recognized criteria of aesthetical presentation, such as general length of lines, number of crossings, and number of bends, are considered the objectives to be minimized. Two algorithms are developed for the stated problem taking into account its specifics. The efficiency of the developed algorithms is evaluated experimentally using randomized test problems of different complexity.

Keywords. Business process diagram, optimization, modeling, orthogonal connectors, business process management

Introduction

The diagrammatic visualization is an important aid in various fields of management and engineering The aesthetic attractiveness is a natural advantage of a drawing. Moreover, according to the general opinion, aesthetical layouts are also more informative and practical [1]. On the other hand the criteria of the aesthetic attractiveness do not always guarantee the informativeness of the diagrams drawn, as it is shown by the experiments with the CASE related diagrams in [15]: "While different generic algorithms, embodying a variety of aesthetics, may produce diagrams that look attractive, a "nice" layout is unlikely to be sufficient for intuitive use". For a discussion on the graph drawing aesthetics we refer to [2], [15], [16]. Although the problem of graph drawing attracts many researchers, and plenty of publications are available, special cases of that problem frequently cannot be solved by straightforward application of the known methods and algorithms. We cite [15] again: "Few algorithms are designed for a specific domain, and there is no guarantee that the aesthetics used for generic layout algorithms will be useful for the visualization of domain-specific diagrams". In [7] aesthetical visualization of aesthetic visualization

[1] Corresponding author

of more specific graphs, business process diagrams, is considered. It is emphasized there, that layout preferences of different user groups can differ essentially, and a set of layout criteria is formulated.

In the present paper we consider a particular problem of the aesthetical drawing of special graphs which are related to business process diagrams of small-medium enterprizes (SME). The algorithms for the aesthetically pleasing visualization of edges of those special graphs are considered, where graphs model business processes, and sequence flows should be visualized assuming the flow objects fixed. Our idea is to reduce the original problem to a problem of the combinatorial multi-objective optimization. For the discussion on the synergy of optimization and visualization we refer to [20]. The developed algorithms are aimed at including into a relatively simple and not expensive software package oriented not only to the consultants of business management but also to the practitioners in SME management [13].

1. Sequence Flow Visualization as a Special Case of Graph Drawing Problem

Our interest in this graph drawing problem is motivated by a request from the developers of a software package for modeling business processes in SME [13]. The latter is oriented to managers and consultants who either design a new SME or search for the possibilities to improve an existing one. The considered business process management methodology is oriented to managers and consultants either designing a new SME or searching for the possibilities to improve an existing one. The Business Process Modeling Notation (BPMN) is accepted as a standard for drawing Business Process Diagrams (BPD) [14]. In the present paper a partial problem of drawing the BPD is considered, namely the problem of drawing the lines which represent the sequence flow for fixed flow objects and defined sequence flow. For the more general problems of constructing BPD we refer, e.g. to [8], [10].

The problem of drawing aesthetical layouts is reformulated as a combinatorial multi-objective problem where the objectives correspond to the criteria usually used to assess the aesthetic attractiveness of a BPD. The developed algorithms can be used interactively when the flow objects are placed by a human user. We are going to continue this research, and subsequently to develop an upper-level algorithm (with respect to the considered in the present paper) for the re-location of the flow objects thus improving the overall aesthetical attractiveness of the considered BPD. The algorithms considered in the present paper will be used by the upper level algorithm as an auxiliary routine for searching Pareto optimal edges for the location of vertices analyzed at upper level.

The problem of drawing the sequence flow is a special case of drawing the edges of a graph where vertices are located on a plane. Moreover, the edges as well as the location of vertices should satisfy some special restrictions. To enable the user to completely understand the information presented by the drawing it normally contains up to 30 flow objects. Therefore in the experiments below we consider graphs with the number of vertices of up to 30. The navigation by the user in BPD is aided by visualization of well perceivable sub-graphs of the considered BPD. The more detailed information concerning the specified flow objects can be extracted by telescoping. For example, a rectangle in the BPD can represent the process, the sub-process, and the task; a process can be decomposed by means of the creation of the child BPD which shows the details of the parent BPD [14]. To denote the flow objects in the considered diagrams, the shapes of three types are used: rectangles, rhombs, and circles which represent the processes, the gateways, and the events correspondingly. The shapes are located in the pool lanes. The sequence flow is represented by the lines constituted of orthogonal segments. BPD can be augmented by data objects and data flows.

In the present paper we focus on the problem of connector routing. Therefore the differences of the flow objects are ignored, and a single rectangular shape is used below to represent the flow objects. Visualization of the graphs, where vertices are drawn as rectangles connected by piecewise vertical and horizontal lines, is commonly used. As the examples, Entity Relationship and UML diagrams can be mentioned among others. Many methods and software implementations of algorithms are available for representing graphs as rectangles connected by orthogonal connectors. However, the immediate application of the available algorithms to the visualization of business processes according to the requirements of the business processes management methodology of [13] is difficult. Of course, basic requirements to the connectors are common for the methods developed to similar problems, and therefore the ideas of known methods were useful in solving our problem.

After a discussion on the general prerequisites related to the orthogonal connectors representing the sequence flow in a business process model, more specific requirements can be stated. The rectangle shapes are located in the centers of a rectangular mesh. The segments of the orthogonal connectors can stretch along borders of the pool lines and in the passages which are orthogonal to these lines, and interpose between the cells of the mesh. An example presented in the Figure 1 illustrates permissible ways for the routing: a connector should join the neighboring vertices represented by small grey circles.

An edge of the graph which models a BPD can be represented by many orthogonal connectors. The abstract criterion of aesthetical image of a BPD depending on the connector can be decomposed into several criteria, and some of these criteria can be evaluated quantitatively. We consider three criteria which seem essential and can be relatively simply evaluated: the total length of connectors, the number of bends, and the number of crossings. All criteria should be minimized. The problem of drawing an aesthetically looking sequence flow is reduced to a multi-objective optimization problem. To the best knowledge of the authors the re-formulation of the initial problem as a multi-objective optimization problem is original although various versions of single-objective optimization problems have been investigated. In the present paper we consider all criteria equally important. However, the relative importance of the criteria can depend on the users of the supposed software package. The relevance of the considered quantitative criteria to the criterion of the subjective perception, and the relative importance of the quantitative criteria are analyzed in [21] by means of a psychological experiment.

2. A Brief Overview of Available Algorithms

The problem of drawing graphs, with the rectangular images for vertices, and with edges composed of pieces of vertical and horizontal lines, is considered in many papers. Depending on the application area in question, the algorithms must satisfy different requirements. Some algorithms, efficient from the point of view of general complexity theory, are described by [19], and [11]. A comprehensive review of algorithms oriented to the

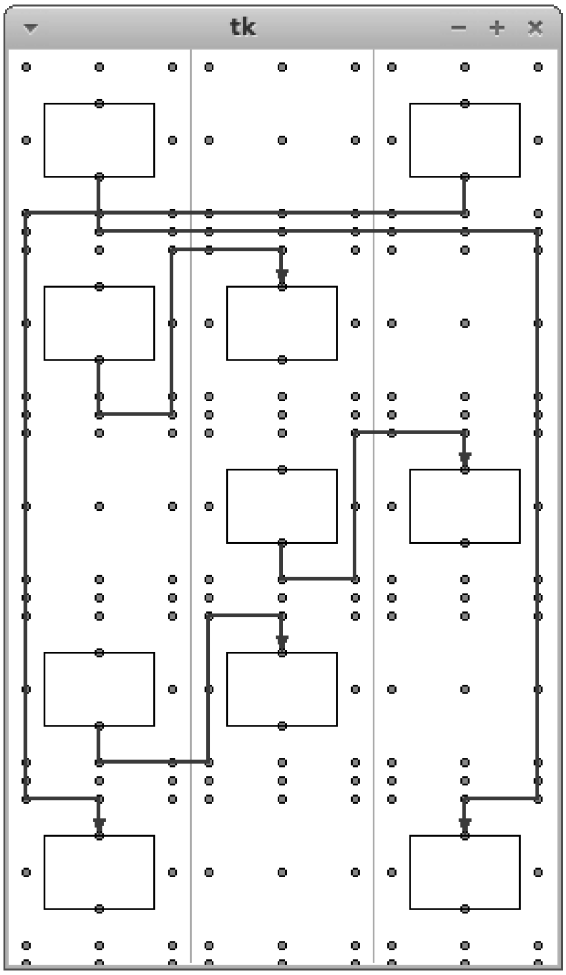

Figure 1. A graphical illustration of a solution to the routing problem

routing of paths for nets on the chip layout to interconnect the pins on the circuit blocks or pads at the chip boundary is presented in [3]. The general purpose routing algorithms are classified in three groups, namely, the maze, line-search, and A*-search groups. Since those algorithms are based on general graph-searching techniques they can be adapted to the specific requirements of both global and detailed routing problems. Different versions of those algorithms are proposed and investigated with the focus on the asymptotic complexity estimates and on the application in the chip design. From the point of view of the BPD drawing, the criteria of aesthetics prevail the criteria important in technological applications emphasized in [3]. In a recent paper by Wybrow et al [18] a brief review of the available algorithms and software, for the construction of orthogonal connectors, is presented from the point of view of the requirements similar to those stated in the previous section. The experimental testing performed by these authors has shown that some of the available software packages, although provide the automatic orthogonal connector routing, produce the routes which may overlap other objects in the diagram.

Popular software packages, Microsoft Visio 2007, and ConceptDraw Pro5, provide the object-avoiding orthogonal connector routing, but in both cases the aesthetic criteria, such as minimizing distance or number of segments, are not taken into account. We cite the conclusion made in the introduction of [18]: "in all current tools that we are aware of, automatic routing of orthogonal connectors uses ad-hoc heuristics that lead to aesthetically unpleasing routes and unpredictable behavior". Agreeing with the latter conclusion as well as with the remarks cited in the Introduction we find the developing of new domain-specific algorithms reasonable.

3. A Modified Shortest Path Algorithm

The most frequently discussed criteria of the aesthetic attractiveness of connectors are length, number of bends, and number of crossings. The first two criteria seem better justified than the last one which seems conditional. Some crossings, e.g. where long edges cross at their middles, is not a negative factor for the perception of the relations indicated by those edges. This objective, however, can be fine tuned to address these corner cases. Having this in mind we start by consider the routing of connectors, focusing on the first two criteria.

Natural candidates for the construction of connectors according to the criterion of connector length are shortest path algorithms. Those algorithms are efficient from the theoretical and practical points of view, and their software implementations were elaborated during intensive, long lasting applications in real world problems. However, the connectors found by a standard shortest path algorithm for the diagrams discussed above frequently are disadvantageous because of many bends. That disadvantage is indeed natural: normally there exist several different paths between two shapes with equal lengths but different number of bends. A trivial solution to find all shortest paths and select one with minimum number of bends is not attractive because of substantial increase of the computation time. Possibly, a new bi-criteria algorithm could be developed for the domain-specific graphs which correspond to the diagrams described above taking into account both criteria – the connector length and the number of bends. However, it would seem not likely to preserve the efficiency of standard shortest path algorithms and software achieved by the many years of refinement. We propose a rather simple modification which enables us to take into account both criteria by using a standard shortest path algorithm.

Let us specify the data for use with a standard version of a shortest path algorithm. The set of vertices of a graph comprises the points denoted in the Figure 1 plus vertices on the shapes marking the ends of the connectors. The set of edges consists of the segments of horizontal and vertical lines between the vertices, if those segments do not cross the shapes. The weights of edges are equal to their lengths. There would usually be several shortest paths in such a graph, and geometrically most of them are of the zig-zag type. This is due to the fact that for such a structure as this graph, the Manhattan distance is used and there are a lot of paths with the same Manhattan distance connecting any two vertices. We also define a modified problem for a graph with the same set of vertices as before but with a different set of edges. The latter also includes the segments of the vertical and horizontal lines with the intermediate points, e.g. in Figure 2 three edges should be considered: (a, b), (b, c), and (a, c). The weight of an edge is equal to the

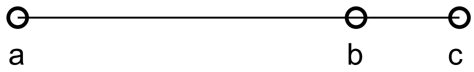

Figure 2. A segment of line represents three edges (a,b), (b,c), and (a,c)

square root of the edge length. In this case, two paths of equal length but comprised of segments of different length can have different weights: the path comprised of small number of long segments will have smaller weight than the paths comprised of large number of short segments. By such a definition of weights, the paths with smaller number of bends implicitly are preferred for being selected by a shortest path algorithm.

The complexity estimates based on asymptotic analysis are not very relevant here since the sizes of problems to be considered cannot be very large otherwise the corresponding diagrams could not be properly surveyed and understood by the user. Nevertheless the complexity estimate is of some interest. Let the size of the orthogonal mesh be $m \times n$, and the number of shapes is denoted by k. The mesh is supposed to be tightly filled: $k = \gamma \times m \times n$, $\gamma < 1$. Assume for simplicity that $m = n$. Then the number of vertices is $O(n^2)$, and the number of edges is $O(n^3)$. Let the shortest path problem in such a graph be solved by Dijkstra's algorithm where the priority queue is implemented as Fibonacci heap. Then the complexity of the algorithm can be estimated as $O(n^3 + n^2 \log(n^2)) = O(k^\alpha),\ \alpha = 3/2$.

The complexity of searching for shortest paths, similar to those searched by the proposed algorithm, can be reduced by taking into account the geometry of the diagram explicitly. However, the gain does not seem counterweighting the loss of flexibility in further modifications of weights on edges taking into account various criteria of the layouts' aesthetics.

4. A Version of the Ant Colony Optimization Algorithm

The modified shortest path algorithm presented in the previous section is primarily oriented to the minimization of the paths' length. In that algorithm, the criterion of bends is taken into account implicitly. The criterion of the number of crossings is not taken into account. The formal involvement of all three criteria by a modification of a known, say the shortest path type, graph algorithm seems difficult. Therefore we start with the general comments on the algorithm selection for a muti-objective optimization problem. The multi-objective algorithms of the type of the classical mathematical programming are efficient for the problems where the objective functions satisfy very restrictive, from the point of view of applications, requirements [12]. Indeed, it seems difficult to find an algorithm of the classical mathematical programming type suitable to the considered problem. As shown in [17], the stochastic algorithms are appropriate for the single objective global optimization of objective functions with various irregularities. The special case of the stochastic algorithms, namely the evolutionary algorithms, are appropriate for solution of various applied multi-objective problems as shown in [5]. Therefore a stochastic metaheuristic algorithm seems appropriate to development of an algorithm for the problem considered. The ant colony optimization (ACO) algorithms are especially oriented to the search for short paths in the complicated graphs [4], [6]. In the recent paper [9] it was shown that the ACO algorithms are efficient in solving the bi-criteria traveling sales person problem. Following the arguments above we have developed a ACO specified for the considered problem. The version of the proposed ant colony optimization algorithm differs from the standard version in the amount of pheromone placed on the path traveled by the ant: it is inverse proportional to the path length, number of bends, and number of crossings squared in contrast to only taking into account the path length. Below we present a description of the algorithm.

1. Mark each edge in the graph with a pheromone value for that edge. In the beginning this value is one. In this way, at the start, all paths are equally likely to be chosen. It is possible to set pheromone values to something other than one at the start. For example we could use some other algorithm, or even ACO itself, maybe with different parameters, to generate a set of paths, and use those paths to modify pheromone values in advance, thus giving the algorithm a head start. Then, the ant colony optimization algorithm could be used to fine tune these paths by, say, emphasizing reduction in the number of bends or crossings.
2. Generate 10 random paths using the procedure outlined below.
   (a) In the beginning the path consists of the starting vertex only.
   (b) Generate successors for the last vertex in path.
   (c) Assign probabilities to each successor by taking pheromone values for edges going from the current vertex to the successor. Normalize these probabilities to add to one.
   (d) Select one of the successors at random according to the probability distribution generated in step (c).
   (e) Attach the selected successor to path.
   (f) If the new vertex is the final one then terminate.
   (g) Go to step (b).
3. For each path run the procedures outlined below.
   (a) For each edge in path add $1/(length + folds + intersections^2)$ to pheromone value of that edge. This is similar to the way that fitness function is computed in the genetic algorithm.
4. For each edge in the graph multiply the pheromone value by 0.9, which simulates pheromone evaporation. In this way the paths that are the shortest, have the least bends and least intersections with other paths will tend to be used most since they will get the highest pheromone values. The other paths will have their pheromone values constantly reduced until it reaches levels so low that almost no ant will choose them.
5. Repeat from step 2 a specified number of times. In the experiments the number of times was set to 500. However values of 200 or even lower were found to be sufficient for the paths to settle.

5. Computational Experiment

The proposed algorithms are implemented in C++, and their performance was evaluated experimentally. Both algorithms are sufficiently fast in the sense that they produce results

Table 1. Mean values and standard deviations of the criteria of solutions found by the modified shortest path algorithm

k	$L(means)$	$L(std)$	$N_b(means)$	$N_b(std)$	$N_c(means)$	$N_c(std)$
6	6.2200	1.9467	6.4800	2.4141	2.0000	1.8257
8	8.1100	2.2514	9.8000	2.7340	3.1000	2.0865
10	10.1500	2.3067	12.8600	3.0385	4.5200	2.4923
12	12.4500	2.8652	16.9400	3.6785	6.8200	3.1120

Table 2. Mean values and standard deviations of the criteria of solutions found by the ants colony optimization algorithm (first mode)

k	$L(means)$	$L(std)$	$N_b(means)$	$N_b(std)$	$N_c(means)$	$N_c(std)$
6	6.8239	0.6681	7.2002	0.7881	1.2925	1.0163
8	8.9890	0.7383	10.3854	0.7362	2.4895	1.4287
10	11.4066	0.9201	13.4010	0.7897	3.9490	1.9956
12	14.2724	1.0055	17.4508	0.7265	6.8063	2.5723

Table 3. Mean values and standard deviations of the criteria of solutions found by the ants colony optimization algorithm (second mode)

k	$L(means)$	$L(std)$	$N_b(means)$	$N_b(std)$	$N_c(means)$	$N_c(std)$
6	7.6706	2.9545	8.9709	3.0115	1.1089	1.2598
8	9.9101	3.3273	12.106	3.3389	2.6572	2.4788
10	12.4737	3.5373	15.1579	3.7099	4.3245	3.7035
12	15.4569	4.3247	19.1217	4.4041	7.7382	5.2840

in time not noticeable by the user. Statistics for the quality criteria of connectors were collected after solving randomly generated test problems.

The modified shortest path algorithm has been applied for the reduced graphs containing only edges at the center of alleys; as can be seen from Figure 1 alleys consist of either three paths or of two paths in our particular problem. It is supposed that the coincident parts of connectors can be separated at the final step of the connectors' refinement, e.g. by the procedure of "nudging" [18].

ACO algorithm has been used in two modes. The first mode corresponds to the reduced graph described in previous paragraph. The second mode corresponds to the original graph.

Locations of shapes were generated at the nodes of the rectangular mesh randomly with uniform distribution. Randomly selected pairs of shapes were connected. The size of the mesh was 3×5. The number of shapes was 6, 8, 10 and 12, representing problems of increasing complexity. While this may seem artificial in the context of BPMN diagram drawing, however we are interested in more abstract properties of the algorithms. The following parameters of the found connectors were evaluated: the total length of connectors (L), the number of bends (N_b), and the number of crossings (N_c). The mean values ($means$) and standard deviations (std) of these parameters were computed using the data of 100 solved problems which were generated randomly as described above. In the case of ACO algorithm, the experiment was repeated 100 times for each of 100 sets of nodes. Means and deviations were then averaged. The results are presented in the Tables.

The experimental results show that both algorithms are of similar efficiency. However the ACO algorithm is more flexible with respect to the increase of number of the criteria considered. Further investigation is supposed including the results of a psychological experiment aimed at the quantitative assessment of the importance of the potentially applicable criteria [21].

Conclusions

Two algorithms of routing of orthogonal connectors, supposed for the aesthetically pleased visualization of edges of special graphs, are proposed. Both of them are sufficiently fast to be useful in the visualization of graphs related to the modeling of the business processes of SME's. The aesthetic criteria of the found connectors are evaluated quantitatively. The experimental results show that the ant colony optimization algorithms are promising for the solution of the considered multi-objective graph optimization problem.

Acknowledgements

The support by Agency for Science, Innovation and Technology (MITA) trough the grant Nr.31V-145 is acknowledged.

References

[1] Battista, G. D., Eades, P., Tamassia, R., Tollis I.G.: *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall (1999).

[2] Bennett, Ch., Ryall, J., Spalteholz, L., Gooch, A.: The Aesthetics of Graph Visualization. In: Cunningham, D. W., Meyer, G., Neumann, L.(eds.) *Computational Aesthetics in Graphics, Visualization, and Imaging* (2007), 1–8.

[3] Chen, H.-Y., and Chang, Y.-W.: Global and Detailed Routing. In: L.-T. Wang, Y.-W. Chang, and K.-T. Cheng, (eds.) *Electronic Design Automation: Synthesis, Verification, and Testing* (ISBN: 0123743648), Elsevier/Morgan Kaufmann (2008).

[4] Colorni, A., Dorigo, M., Maniezzo, V.: *Distributed Optimization by Ant Colonies, actes de la premiere conference europeenne sur la vie artificielle*, Elsevier Publishing, Paris, France (1991), 134–142.

[5] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons (2009).

[6] Dorigo, M., Birattari, M., Stutzle, T.: Ant Colony Optimization, Technical Report No. TR/IRIDIA/2006-023 (2006).

[7] Effinger, Ph., Jogsch, N., Seiz, S.: *On a Study of Layout Aesthetics for Business Process Models Using BPMN*, Lecture Notes in Business Information Processing, Vol. 67 (2010), 31–45.

[8] Effinger, Ph., Kaufmann, M., Siebenhaller, M.: *Enhancing Visualizations of Business Processes*, Lecture Notes in Computer Science, Vol. 5417 (2009), 437–438.

[9] C. Garcia-Martinez, O. Cordon, F. Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operations Research* **180** (2007), 116–148.

[10] Kitzmann, I., König, Ch., Lübke, D., Singer, L.: A Simple Algorithm for Automatic Layout of BPMN Processes, In: *1st International Workshop on BPMN (CEC-09 – 11th IEEE Conference on Commerce and Enterprise Computing)* (2009), 391–398.

[11] Lee D., Yang C., Wong C.: Rectilinear paths among rectilinear obstacles. *Discrete Applied Mathematics* **70**(3) (1996), 185–216.

[12] K. Miettinen. *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers (1999).

[13] ORGSOFT, http://www.orgsoft.lt.

[14] Owen M., R.Jog.: BPMN and Business Process Management (2003), 1–27, http://www.bpmn.org.

[15] Purchase, H. C., McGill, M., Colpoys, L., Carrington, D.: Graph drawing aesthetics and the comprehension of UML class diagrams: an empirical study. In: *Australian Symposium on Information Visualization, Conferences in Research and Practice in Information Technology*, Vol. 9 (2001).

[16] Purchase, H. C.: Metrics for graph drawing aesthetics. *Journal of Visual Languages and Computing* **13**(5) (2002), 501–516.

[17] A. Törn, A. Žilinskas. *Global optimization*, Lecture Notes in Computer Science, Vol. 350 (1989), 1–255.

[18] Wybrow M., Marriott K., Stuckey P.: *Orthogonal Connector Routing*. Lecture Notes in Computer Science, Vol. 5849 (2010), 219–231.

[19] Yang C-D., Lee D., Wong C.: Rectilinear Path Problems among Rectilinear Obstacles Revisited, *SIAM J. Comput.* **24** (1995), 457–472.

[20] Žilinskas A., Žilinskas J.: Optimization based visualization, In: C.Floudas and P.Pardalos (Eds.), *Encyclopedia of Optimization*, Springer (2009), 2785–2791.

[21] Žilinskas A., Mackutė-Varoneckienė, A., Varoneckas A.: Weighing Criteria of Aesthetic Attractiveness of Layouts of Business Process Diagrams, In: *Proceedings of STOPROG 2012*, in print.