I T H E A

International Journal
INFORMATION THEORIES & APPLICATIONS

2016 Volume 23 Number 4

International Journal

INFORMATION THEORIES & APPLICATIONS
Volume 23 / 2016, Number 4

Editorial board
Editor in chief: **Krassimir Markov** (Bulgaria)

Alberto Arteta	(Spain)	**Luis F. de Mingo**	(Spain)
Aleksey Voloshin	(Ukraine)	**Lyudmila Lyadova**	(Russia)
Alexander Eremeev	(Russia)	**Martin P. Mintchev**	(Canada)
Alexander Kleshchev	(Russia)	**Natalia Bilous**	(Ukraine)
Alexander Palagin	(Ukraine)	**Natalia Pankratova**	(Ukraine)
Alfredo Milani	(Italy)	**Rumyana Kirkova**	(Bulgaria)
Avram Eskenazi	(Bulgaria)	**Stoyan Poryazov**	(Bulgaria)
Boris Fedunov	(Russia)	**Tatyana Gavrilova**	(Russia)
Constantine Gaindric	(Moldavia)	**Tea Munjishvili**	(Georgia)
Galina Rybina	(Russia)	**Valeriya Gribova**	(Russia)
Hasmik Sahakyan	(Armenia)	**Vasil Sgurev**	(Bulgaria)
Ilia Mitov	(Bulgaria)	**Vitalii Velychko**	(Ukraine)
Juan Castellanos	(Spain)	**Vitaliy Lozovskiy**	(Ukraine)
Koen Vanhoof	(Belgium)	**Vladimir Donchenko**	(Ukraine)
Krassimira B. Ivanova	(Bulgaria)	**Vladimir Jotsov**	(Bulgaria)
Leonid Hulianytskyi	(Ukraine)	**Vladimir Ryazanov**	(Russia)
Levon Aslanyan	(Armenia)	**Yevgeniy Bodyanskiy**	(Ukraine)

International Journal "INFORMATION THEORIES & APPLICATIONS" (IJ ITA)
is official publisher of the scientific papers of the members of
the ITHEA International Scientific Society

IJ ITA welcomes scientific papers connected with any information theory or its application.
IJ ITA rules for preparing the manuscripts are compulsory.
The **rules for the papers** for IJ ITA are given on *www.ithea.org*.

Responsibility for papers published in IJ ITA belongs to authors.

International Journal "INFORMATION THEORIES & APPLICATIONS" Vol. 23, Number 4, 2016

Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with:
Institute of Mathematics and Informatics, BAS, Bulgaria,
V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
Universidad Politécnica de Madrid, Spain,
Hasselt University, Belgium,
St. Petersburg Institute of Informatics, RAS, Russia,
Institute for Informatics and Automation Problems, NAS of the Republic of Armenia

Printed in Bulgaria
Publisher ITHEA®
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com
Technical editor: **Ina Markova**

Copyright © 2016 All rights reserved for the publisher and all authors.
® 1993-2016 "Information Theories and Applications" is a trademark of ITHEA®
® ITHEA is a registered trade mark of FOI-Commerce Co.

ISSN 1310-0513 (printed) **ISSN 1313-0463 (online)**

DECISION-MAKING IN GROUPS OF INTERVAL ALTERNATIVES

Gennady Shepelev

Abstract: *Problem of comparing alternatives with numerical quality indicators, which due to uncertainty have interval representations, is considered. It is demonstrated inevitability of presence in similar tasks of irremovable risk of making a wrong decision on preference of alternatives. It is emphasized necessity of the account in dealing with such problems both indicators of preference for analyzed alternative in comparison with other objects in their group and risk indicators of making wrong decision. Advantages and disadvantages of available methods of interval alternatives comparing are analyzed. Namely, methods of comparison are considered based on principles of dominance by probability, evaluation methods based on average indicators, new method of collective risk estimating, and methods of "mean – risk" approach. The arguments are given that methods of dominance by probability and methods based on average indicators as tools, which do not allow estimate risk of making a wrong decision explicitly, can play only an auxiliary role in decision-making. Advantages of the method of collective risk estimating consist in account of integral risk in concrete group of compared interval alternatives. Disadvantages are consequence of the fact that the method compares only relative preferences of alternative and permits take into account acceptability of separate alternative only after additional constructions. In the framework of "mean – risk" approach properties of such convenient for practice risk indicator as average semideviation were studied for some simple but important for applications distribution functions of preference chances. It is shown that in some cases instead of average semideviation indicator using of indicators of risk based on results of methods evaluating of deviation from target marks is more expedient. An advantage of methods of "mean – risk" approach is the possibility of calculating for each alternative pair of basic indicators needed for the evaluation interval alternatives, indicator of acceptability of alternatives and indicator of associated risk. The disadvantages include the fact that both of these indicators are calculated for an alternative as an independent object, which is not associated with others in compared group. Comparison of the alternatives on preference is based on values of these indicators but dependence of the risk on the context is not taken into account. Due to presence of both advantages and disadvantages of the methods is proposed to use a three-step approach to decision-making on the selection of interval alternatives, which is based on consistent using of different methods: A stage of the preliminary analysis of alternatives for the purpose of culling some of their number and reducing the dimension of the problem and size of collective risk by this is suggested to include in the procedure of decision making on the choice of the preferred objects. Method* *of average estimates is suggested to use at this stage. Using of the collective risk estimating techniques with isolating of a small number of alternatives as acceptable by the indicator of preference, as well as having the lowest risk of wrong estimates, is recommended at the following stage and methods of "mean – risk " approach at the final stage.*

***Keywords**: comparing interval alternatives, methods of probabilistic dominance, collective risk estimating, "mean-risk" methods, joint using of the methods.*

***ACM Classification Keywords**: H.1.2 Human information processing. G3 Distribution functions. I.2.3 Uncertainty, "fuzzy", and probabilistic reasoning.*

Introduction

Let us call by interval alternatives such objects of comparison or evaluation whose quality indicators that are measured in quantitative scales have, due to uncertainty, interval representations. Such representation of numerical indicators arises, for example, in forecasting problems in attempts to predict the future values of point, by their nature, quantities. It is assumed that interval estimate includes all possible, up to the available knowledge, point implementations of forecast quantity. But in the future, when the uncertainty is removed, this quantity will receive certain the only numerical value.

The objective of comparing two or more alternatives is selection of one or more preferred ("best") in a sense objects. The purpose of evaluation is comparison of quality indicator of interval alternative with some preassigned point benchmark that characterizes acceptable for assessing subject threshold value of the quality indicator. The criteria of comparison or evaluation, although have relations with the indicators of quality by their domains of definition, are not necessarily expressed in terms of the values of the lasts and may have dimensions, which are different from the dimensions of quality indicators. Further we will assume for definiteness that such situations take place when the greater values of the quality indicators are preferable than small values.

Problems of comparing and evaluation of interval alternatives due to their characteristics cannot be exhaustively solved by purely mathematical methods. Indeed in the case comparing of alternatives for general configurations, when the compared intervals have non-zero intersection, in principle one cannot with certainty conclude, which alternative in their entirety will be preferable. Any alternative may be "better" in the future, at the time of "removal" of uncertainty, when the interval estimates are replaced by exact (point) values of quality indicator. Therefore at the time of the comparison can be judged only on the chances that one alternative will be preferable to others. Therefore there is always a risk that in fact namely another alternative but not tested would be better. In problems estimating of the acceptability of separate alternative point benchmark, with which the interval quantity is compared, divides the interval object into two areas. All point implementations that suits to decision-maker (DM) are located in one area, those implementations, which are unacceptable to DM, lie in another area. Again one can only judge about chances of getting the point implementations in the desired area and the risk of them misses in it.

Thus formal methods of comparison and estimation serve here only as a tool of information-analytical support for the decision making process and cannot guarantee choice of truly the best object in the process of comparing or estimating. Thus such problems are problems of the decision-making theory as in the decision-making process have to involve preferences of DMs or experts and take account of their risk tolerance. So adequate to essence of the problem comparison should take into account at least two criteria, measure of preference of alternative in relation to others in their totality and risk measure. One can note that this requirement is not always fulfilled both in practice and in the recommendations on the use of comparison methods. Similarly indicator of the acceptability of alternative and indicator of risk that result of estimating will be in fact wrong must be taken into account during estimating. Human involvement in the decision-making process can determine not only the choice of the tools of describing the uncertainty but also, due to previous experience and knowledge of human being, the choice of the comparison and estimation methods that lead to the result indicators, which are familiar to the expert or DMs. Each of the available methods of comparison and evaluation allows to calculate its measures of alternative preference in relation to the other alternatives in their totality as well as indicators of the acceptability of alternatives during estimating and (but not always) their risk measures. For some configurations, i.e. relative location, of compared alternatives and types of uncertainty describing predictions of different methods are equivalent for others not. In this regard it should be stated that at present there is no approach transcending all others in quality of recommendations obtained on its basis. Each method has its advantages and disadvantages. Joint using of different methods on various successive stages of the decision-making process is probably the best way to combine the power of formal methods and knowledge of experts and DMs. Decision-making for comparing and estimating of interval alternatives remains both science and art, as it was before [Larichev, 1979].

The purpose of this paper is to compare different approaches and methods of evaluation and comparing of interval alternatives and identify their place in the decision-making process of the selection of preferred objects. We start with a matching of interval alternatives comparison methods.

Methods of interval alternatives comparison

We assume that all the alternatives are comparable in preference (i.e. system of alternatives is full). Disjunction containing comparable interval alternatives is not rigorous in this case that is such that only one object can be chosen as preferred. This choice depends now on the chances of preferences of the disjunction members. Similar relations of disjunction members are based on the degree of assurance in the truth of the hypotheses about preference an alternative from their totality, which is tested by DM. Such relations may be called by relations including the risk. Let us assume that from all possibilities of uncertainty description the tools of distribution functions, similar to apparatus that used in the theory of probability, is selected here for the quantification of preference chances for compared interval alternatives or subsets of values contained in them. This apparatus is the most familiar, in our experience, expert practitioners. It is important as expert analysis of practical problems is most productive when it is conducted in the usual for domain experts’ language with using terminology understandable to them [Petrovsky, 1996].

Two main types of problems distinguish in the theory of alternatives comparing under uncertainty. These are of unique choice and problems of repeated choice. Each of these problems types requires for the analysis the specific comparison and risk criteria. So value of mathematical expectation of quality indicator as random variable is adequate criterion for problems of repetitive choice. The problems of forecasting deal mainly with situations of unique choice. This requires, in general, rejection of average estimates, or, if they are used, the mandatory accounting as no less important criterion estimations of risk calculated on the basis of certain indicators.

Let’s note In addition to the above that the available methods of interval alternatives comparison compare them in different ways. In the methods of stochastic dominance pairwise comparison of alternatives carried out only on the basis of the behavior of the distribution functions defined on intervals-carriers, without taking into account the numerical characteristics of the firsts [Levy, 2006].

In the “mean – risk” approach [Fishburn, 1977] compared alternatives are considered as isolated, not “interactive” objects. Value of preference criterion is calculated for each alternative separately and then, regardless of this indicator, a risk indicator is computed again separately for each compared object. The problem of comparing is solved then as a two-criterion task. Mathematical expectation value is here the criterion of preference both for problems of unique choice and repeated choice. Such indicators as variance, left and right semivariances, left and right mean semideviations and the others act as risk criteria [Baker, 2015]. Let draw attention to the fact that the calculated values of the comparison criteria for these methods do not depend on the number of alternatives to be analyzed in their group.

In the method of collective risk estimating [Shepelev, 2015], when produced direct calculations of preference chances of alternatives in comparison with others in their totality, compared objects are viewed as interconnected community. Because apparatus of distribution functions was selected for quantification of preference chances and associated risks, the problem of comparing can be analyzed in the framework of probability logic approach [Nilsson, 1986]. In accordance with this approach in addition to the truth or falsity of logical statements intermediate logical values are possible. They are interpreted as chances of truth. The use of this approach to interval alternatives comparing allows calculating both the chances of alternative preferences and associated risks. Risk of choice of an alternative in their group as the preferred depends on the relative position of alternatives (configuration of alternatives) and on the number of compared objects. An original interaction of compared objects leads to a collective effect [Shepelev, 2015; Sternin, 2015], which consists in the fact that the properties of objects of interacting components of the system is significantly different from those of relatively independent objects. Therefore risk of making the wrong decision when choosing a preferred object increases with the growing number of compared alternatives. It seems that this approach is more consistent with the essence of problems of unique choice.

Let note that we deliberately leave aside the methods based on the construction of utility functions. These techniques allow us, on the one hand, to find point estimates, equivalent to the compared interval estimates (“deterministic equivalents”) [Keeney, 1993], that is undoubtedly attractive for practitioners, but on the other hand requires detailed information about the preferences of the DM, her /his risk appetite and the use of complicated procedure to transform this information into specific utility functions. Allowing to calculate preference indicators of interval alternatives in their compared pair [Shepelyov, 2012], methods of comparing based on the construction of utility take into account the risk of making the wrong decision only indirectly during including the DM's preferences and propensity for risk in the specification of the parameters of the selected class of utility functions.

Let us consider now the above methods of comparing and analyze their possible place in the process of decision-making concerning choosing of a preferred alternative. Let start with the methods of stochastic dominance namely stochastic first-order dominance (or probabilistic dominance, dominance according to “chances" in line with the terminology adopted above) as having simple meaningful sense.

Methods of probability dominance

They say that the interval alternative I_1, where distribution F_1 of the random variable X_1 is given, dominates alternative I_2, where distribution F_2 of the random variable X_2 is given, if for a set of possible point implementations $I_1 \cup I_2$ for any point implementation x chances $F_1(X_1 < x)$ are not more than chances $F_2(X_2 < x)$, and at least for one point implementation they are smaller. In other words the graph of the distribution function F_1 for alternative I_1 lies always below the graph of the distribution function F_2, possibly coinciding with the first in some parts.

Keeping in mind our objectives we will analyze this method in more detail for the two distributions of chances, uniform and triangular ones. The first of them is consistent with the principle of maximum by Gibbs - Jaynes, the second is a visual model of common among practitioners unimodal distributions.

We will define interval objects $[L, R]$ by the left L and right R boundaries, $L < R$. From the four, up to permutations, different configurations of compared alternatives pairs, coinciding intervals; intervals without intersection; configurations of right shift and embedded intervals let focus on discussing the last two configurations, which have the greatest interest for the real-life problems.

In the case of right shift configurations, when $L_2 < L_1 < R_2 < R_1$, for uniform distributions alternative I_1 dominates alternative I_2 by probability. Indeed if by definition $\Delta I_i = R_i - L_i$, $i = 1, 2$ then

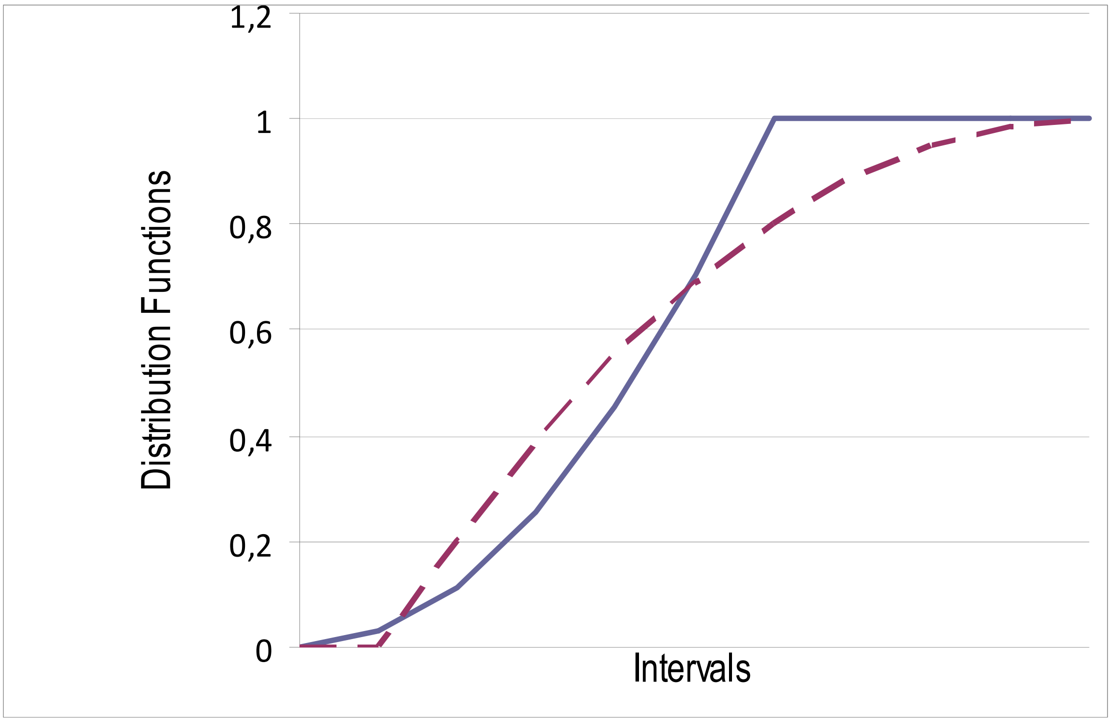
$$F_i(x) = \begin{cases} 0, x \le L_i \\ \dfrac{x - L_i}{\Delta I_i}, L_i < x < R_i \\ 1, x \ge R_i \end{cases} \quad (1)$$

When $\Delta I_1 = \Delta I_2$ line segments, which represent graphs of distribution functions F_i, are parallel, in other cases they are intersected at a point $I_{in} = (L_2\Delta I_1 - L_1\Delta I_2)/(L_2\Delta I_1 - L_1\Delta I_2)$. One can verify that the inequality $L_1 < I_{in} < R_2$ is not met for the right shift configurations, and therefore in area $[L_2, R_1]$ $F_1 \le F_2$. Therefore the first alternative dominates the second by probability[1].

This conclusion is valid for any value of the uncertainty zone $[L_1, R_2]$ for point implementations of interval alternatives. Does it mean that DM should always select as the preferred first alternative due to the dominance of the second by probability? It seems that not because following this requirement means the neglect of the risk of making a wrong decision on the preference. DM can but should not make such a choice. We also call attention to the fact that after choosing one of the alternatives on base of the results of the analyzed approach DM still has not an estimate of the acceptability of the alternative. DM received only estimate that one alternative is preferable to another.

Will demonstrated above dominance by probability to have place for distributions other than the uniform? Answer is negative if distributions are strongly skewed in opposite directions (have "long tails", which are pointing in different directions). It is shown on Figure 1, with using the relations similar to (1), for the functions of triangular distributions with modes $M_1 = 2.1$; $M_2 = 6.9$ for intervals $I_1 = [2, 11]$, $I_2 = [1, 7]$.

[1] Note that in this case I_1 is more preferable than I_2 also according to frequently used criterion of mathematical expectation.



Legend: - - - for I_1; — for I_2.

Fig. 1. An example of absence of dominance by probability for the right shift case

Turn now to the case of embedded intervals. Let I_2 embedded in I_1. The intersection point I_{in} of distribution functions lies for uniform distribution in the interval $[L_2, R_2]$. And what is more I_2 dominates I_1 by the probability in the area lying to the left of I_{in} and on the contrary to the right of this point.

One can see that $F(I_{in}) = F_1(I_{in}) = F_2(I_{in}) = (L_2 - L_1)/(\Delta I_1 - \Delta I_2)$. Length of the region where I_2 dominates I_1 is more than length of the region where I_1 dominates I_2 if $F(I_{in}) < ½$, that is if $R_1 + L_1 > R_2 + L_2$.

Thus for this configuration has place a certain harmonization of results based on comparing criterion of dominance by probability and mathematical expectation criterion. However this harmonization is expressed less clearly than for the right shift configurations.

In closing the discussion of this approach note that use of the dominance by probability principle to eliminate certain alternatives from their set for decreasing their number to reduce collective risk is problematic. In the framework of this approach there is only pairwise comparison of alternatives, the comparison results do not include risk estimates explicitly, there are no estimates of the acceptability of separate alternative.

Methods of collective risk estimating

In the framework of method of collective risk estimating compared alternatives are considered as an interconnected totality. Comparison is carried out here in general, "as a whole". Risk of choosing an alternative as preferred in their set depends here not only on configuration of compared alternatives, but also on their amount in the set [Shepelev, 2015]. The presence of the group of mutually influencing alternatives increases the risk of making the wrong decision during choosing a preferred alternative. This is due to a "collective effect" just as it happens, for example, in condensed matter physics when properties of condensed matter systems composed of interacting components significantly differ from properties of more or less independent parts [Halperin, 2010].

Dimensionless chances of truth of tested by expert hypothesis on preference of an analyzed alternative relative to others are comparison criteria within this approach. Chances of truth of the opposite hypothesis, which complement the first chances up to unity, serve as a measure of risk. In this approach point implementations of different alternatives from analyzed set are considered as independent and priori all the alternatives have equal rights with respect to the choice.

The start step in the realization of this approach is pairwise comparing alternatives, when the number of objects to be compared and its impact on risk do not take into account [Shepelyov, 2013; Shepelev, 2014]. Criterion of comparison of interval alternatives with arbitrary distributions of chances, which was called "assurance factor", was proposed on this way. It is equal to the difference between chances of the truth of tested hypothesis on preference of an alternative in their set and the chances of the truth of the opposite hypothesis. Numerical (for arbitrary distributions of chance) and analytical (for uniform and triangular distributions) methods calculating the assurance factor as well as decision-making procedures based on this criterion were proposed. The assurance factor and chances of preference are equivalent as comparison criteria. The first of them is more convenient in some cases. For example, for some simple distributions of chances one may establish a relation between such criteria as the difference of the averages for two compared alternatives and assurance factor [Shepelev, 2014] and their dissimilarity in other cases.

Let give a brief overview of the results obtained in the framework of the method of collective risk estimating in paper [Shepelev, 2015].

Suppose that there are K alternatives I_i, i = 1, 2,…, K with the same interval quality indicators. Let dimensionless quantity $C(I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},…, I_K))$ is the chances, in other words degree of assurance, in the truth of a testable hypothesis of preference, that the alternative I_i is more preferable than all at once alternatives ($I_1, I_2,.., I_{i-1}, I_{i+1},…, I_K$) from initially given their set (I_i is "better" of others "on the whole"). The term "all at once" means here that

$$I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_K) \equiv (I_i \succ I_1) \wedge (I_i \succ I_2) \wedge (I_i \succ I_3) \wedge \dots \wedge (I_i \succ I_{i+1}) \wedge \dots \wedge (I_i \succ I_K),$$

where $\equiv$ and $\wedge$ are symbols of equivalence and conjunction respectively.

Risk that I_i would not preferred in reality will be measured by means of dimensionless quantity $R_s(I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_K))$ complementing previous chances to unity so that

$$Rs(I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_K)) = 1 - C(I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_K)).$$

As can be seen $R_s(I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_K))$ is degree of assurance in the truth of a hypothesis, which is opposite to the testable hypothesis of preference.

Equivalently $R_s(I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_K)) = C(\neg (I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_K)))$, where $\neg$ is symbol of negation. It is shown in [Shepelev, 2015] that the following relations hold for chances

$$C(I_1 \succ (I_2, I_3,\dots, I_K)) + C(I_2 \succ (I_1, I_3,\dots, I_K)) + C(I_3 \succ (I_1, I_2, I_4,\dots, I_K)) + \dots + C(I_K \succ (I_1, I_2,\dots, I_{K-1})) = 1 \quad (2)$$

and for risks

$$Rs(I_1 \succ (I_2, I_3,\dots, I_K)) + Rs(I_2 \succ (I_1, I_3,\dots, I_K)) + Rs(I_3 \succ (I_1, I_2, I_4,\dots, I_K)) + \dots + Rs(I_K \succ (I_1, I_2,\dots, I_{K-1})) = K - 1.$$

Let us prove the relation (2). The statement $\neg (I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_K))$ means that at least one alternative from their compared set would be preferable than I_i. Let illustrate the meaning of introduced in this way the measure of risk for case of three alternatives. Here we have the following possible preferences and chains of disjunctions:

$$((I_1 \succ I_2 \succ I_3) \vee (I_1 \succ I_3 \succ I_2)) \vee ((I_2 \succ I_1 \succ I_3) \vee (I_2 \succ I_3 \succ I_1)) \vee ((I_3 \succ I_1 \succ I_2) \vee ((I_3 \succ I_1 \succ I_2))$$
$$\equiv (I_1 \succ (I_2, I_3)) \vee (I_2 \succ (I_1, I_3)) \vee (I_3 \succ (I_1, I_2)) \text{ or } Rs(I_1 \succ (I_2, I_3)) = C(I_2 \succ (I_1, I_3)) + C(I_3 \succ (I_1, I_2)).$$

Hence for K alternatives we have (2).

One can see now that $C(I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},.., I_K))$ is monotonically non-increasing function of K, that is the chances that a certain alternative would be preferable in comparison with all the others do not increase with increasing number of the alternatives. Indeed, if the number of compared alternatives is increased the number of non-negative terms in the unit sum of corresponding chances in (2) is also increased. Therefore

$$C(I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_K)) \leq C(I_i \succ (I_1, I_2,.., I_{i-1}, I_{i+1},\dots, I_{K-1})),$$

Then corresponding risk will be monotonically non-decreasing function of number of compared alternatives:

$$Rs(I_i \succ (I_1, I_2,\ldots, I_{i-1}, I_{i+1},\ldots, I_{K-1})) \le Rs(I_i \succ (I_1, I_2,\ldots, I_{i-1}, I_{i+1},\ldots, I_K)).$$

By another words the more of number of the alternatives the more risk of wrong decision-making.

The effect of the comparison of interval alternatives "as a whole" is manifested primarily in reducing value of preference chances for each alternative with respect to its chances under pair-wise comparison. This leads to a quantitative increasing risk value of selection as preferred alternative such one, which may not actually be per se later. As already noted the nature of this effect lies in the fact that in the presence of non-zero intersection for already two compared alternatives there is a non-zero risk of making the wrong decision. This risk is enhanced with increasing amounts of compared alternatives especially if some of these chances are not too different from each other. However, it should be borne in mind that the perception of risk is individual and can vary from one DM to another. Therefore the risk value resulting from the use of the proposed method is nothing more than a calculated risk, which can serve only as an estimate for the DMs.

What can be done to reduce the calculated risk? During deciding on preferred alternative choice or in the process of ordering alternatives by preference it's useful to conduct a preliminary analysis of their initial set. Firstly, after selecting an alternative that preference is tested, one should select in the set of alternatives those, which do not have the intersection with analyzed alternative. If the left boundary of such intervals no less than the right boundary of the tested one the latter is certainly worse. If the right boundary of such intervals not greater than the left boundary of the tested one they can be excluded because they are certainly worse than the last interval. Secondly, one may try to unify some similar or complementary alternatives. By reducing the number of intervals in their initial set one may increase the calculated preference chances of analyzed alternative and decrease risks. At last, after calculating the preference chances of tested alternative during pairwise comparisons it is advisable to exclude those alternatives whose preference chances with respect to tested alternative is less than 0.5 and, respectively, the risk is more than 0.5. Other possibilities to reduce the number of comparable alternatives and to decrease by that the collective risk discussed below in the discussion of using "mean-risk" methods.

Are there any other amendments to the results of the pairwise comparison of alternatives due to "collective" effect? Particularly important is the following question: is there difference of the alternatives order in their set defined by the "collective" preference chances and the order for pairwise comparison? The answer to this question is negative: the order established in the process of pairwise comparison is the same as the order for comparison "as a whole".

Thus for this approach the “best” alternative will be the alternative with the highest chances at pairwise comparisons in the set of compared alternatives. However adequate the risk estimation of making wrong decision we obtain by comparing this alternative simultaneously with all the others, “as a whole”.

Advantages of this approach consist in account of integral risk in a concrete group of compared interval alternatives that permit to DM to get an idea of the true size of surprises lie in wait for him. Other methods of comparing do not permit do so. Disadvantages are a consequence of the fact that the method compares only relative preferences of alternative and does not take into account acceptability of separate alternative.

“Mean — risk” methods of comparing

“Mean-risk” methods are free from these disadvantages and complement method of collective risk estimating. Let us now consider “mean-risk” methods in a version, where the preference criterion is value of mathematical expectation and risk criterion, in accordance with the recommendations of the papers [Ogryczak, 1999; Grechuk, 2012], is the mean absolute semideviation S_I. It is natural to suppose that one alternative is more preferred than another in their compared pair if this alternative is better according to the one of the two criteria and no worse according to the second criterion.

As a measure of alternative preference (as well as of an acceptability) is here mathematical expectation E of a random variable X defined on I, then the measures of the average deviation of the random variable on the left S_{Il} and on the right S_{Ir} from E (“semideviations”) are good indicators of risk. Conveniently they have the same dimension as that of E and besides, as it can be shown, S_{Il} = S_{Ir}.

By the definition

$$S_{Il} = \int_L^E (E-x)f(x)dx,\ S_{Ir} = \int_E^R (x-E)f(x)dx$$

because

$$\int_L^R (E-x)f(x)dx = \int_L^E (E-x)f(x)dx - \int_E^R (x-E)f(x)dx = 0,$$

so S_{Il} = S_{Ir} = S_I.

Let consider the behavior of S_I for uniform and triangular distributions in more detail. For uniform distribution have: $E_U = (L + R)/2$, and $S_{IU} = (R – L)/8$.

In the case of triangular distribution with mode M its density consists of two branches, the left f_l lying on the graph of the density on the left of M, $f_l = 2(x – L)/[(R – L)(M – L)]$ and the right, $f_r = 2(R – x)/[(R – L)(R – M)]$. Mathematical expectation equals $E_T = (R + M + L)/3$. There are two possibilities for location of the mode: $M \le E_T$ and $M > E_T$. One can show that if $M > E_T$ then $M > (L + R)/2 = E_U$, if $M \le E_T$, then $M \le E_U$.

For $M \le E_T$ have:

$$S_I = S_{Ir} = \int_{E_T}^{R} (x - E) f_r(x) = \frac{(R - E_T)^3}{3(R-L)(R-M)} = \frac{(2R - L - M)^3}{81(R-L)(R-M)}.$$

And for $M > E_T$:

$$S_I = S_{Il} = \int_{L}^{E_T} (E - x) f_l(x) = \frac{(E_T - L)^3}{3(R-L)(M-L)} = \frac{(R + M - 2L)^3}{81(R-L)(M-L)}.$$

One can see that as a function of mode M semideviation $S_I(M)$ is convex downward, monotonically decreasing on the interval $[L, (L + R)/2]$ and monotonically increasing on the interval $[(L + R)/2, R]$ function. The function is symmetrical about a vertical axis $M = (L + R)/2 = E_U$, its minimum S_{Imin} is attained at the point $M = E_U$, $S_{Imin} = (R – – L)/12$ and maximum S_{Imax} at points $M = L$ and $M = R$, $S_{Imax} = S_I(L)= S_I(R) = 8(R – L)/81$.

It is useful for experts to take into account the following during selection of chances distribution functions to describe their knowledge about the interval indicators of quality. The transition from uniform distributions to the others, for example, triangular, means availability of more knowledge about the object. This reduces the risk indicator value and therefore $S_{Imax} < S_{IU}$ (for the same carrier interval). Choice of the mode, which equals to the mean of the corresponding uniform distribution E_U, in triangular distributions results in the lowest risk indicator value and deviations from the values of M on both sides from E_U leads to a symmetric growth of risk indicator. However, these deviations from the viewpoint of the alternatives comparison are not equivalent. Deviations of M from E_U on the right lead to increasing of expectation value E_T, i.e. to increasing of preference indicator in “mean – risk” approach, and on the left to decreasing of E_T.

The indicator S_{Il} characterizes possible, because of the uncertainty, the average unfavorable deviation of alternative quality indicator from the value of the mathematical expectation. Course the average deviation is significantly less than the maximum possible deviation. So S_{IU} is four times less than the highest possible negative deviation. However the choice of the mathematical expectation as a preference indicator may not reflect the preferences of DM and her/his risk appetite. So for uniform distribution of the chances for such a choice point exactly half of the possible implementations of the quality indicator are less than the value of mathematical expectation.

It is advisable therefore the introduction of a risk measure $S_{II}(a)$, which is similar to the average semideviation but is associated with any valid target value *a* of an alternative quality indicator:

$$S_{II}(a) = \int_{L}^{a} (a-x) f(x) dx .$$

Consider some of the properties of this indicator for the simplest case of uniform distributions of chances.

Let α (0 < α < 1) is the desired for DM level of chances that the point implementations of the quality indicator will be less than *a*. It is clear that lower values of α are more desirable but such values may be connected with undesirable values of target indicator.

For uniform distributions $a = (1 - \alpha)L + \alpha R$. DM can now either appoint the value of α and determine *a* or do the opposite. For $S_{II}(a)$ can obtain the following relationship: $S_{II}(a) = (a - L)^2/[2(R - L)] = S_{II}(\alpha) = \alpha^2(R - L)/2$. Coefficient $\alpha^2/2$ can be called a risk factor K_r. By definition $K_r = \alpha^2/2$. It is easy to see that the smaller value of the target indicator the smaller the value of risk factor.

In real problems $R > 0$, $L < 0$ because for $R < 0$ alternative should be excluded from the number of compared and for $L > 0$ alternative most often is a priori suitable for the realization. Under such conditions $S_{II}(\alpha) > 0$ but the negative values of the expectation *E* are possible. Let require that only alternatives with $E > 0$ be permitted to the comparison; i.e. such that $R > -L$. We also require that for suitable to comparison alternatives values of the risk indicator were below the target indicators: $S_{II}(\alpha) < a$. This gives for suitable alternatives the condition: $R > - L(2 - - 2\alpha + \alpha^2)/[\alpha(2 - \alpha)]$. The coefficient $K_s = (2 - 2\alpha + \alpha^2)/[\alpha(2 - \alpha)]$ may be called the suitability factor.

Naturally to require also that the selected by DM value of the target mark was positive. This requirement leads to the following condition for *R*: $R > -[(1 - \alpha)/\alpha]L$. Coefficient $K_p = (1 - \alpha)/\alpha$ can be called a factor of positivity (for *a*). Values of chances α, for which the positivity condition is not met, should not be tested by DM.

One can be shown that $K_s > K_p$, since, besides, $K_s > 1$, the condition of the suitability of alternatives to comparison $R > - K_sL$ provides simultaneous fulfillment of the requirements $E > 0$, $a > 0$ and $S_{II}(\alpha) < a$. Thus alternatives with negative left boundaries for which $R < - K_sL$ may be excluded.

Let calculate values of the risk, suitability and positivity factors for chances α that less than 0.5, namely such chances most likely will be tested by DMs. Results of the calculations are shown in Table 1.

Table 1. Some risk, suitability and positive factors values

Chances (α)	0.1	0.2	0.3	0.4	0.5
Risk Factor	0.005	0.02	0.045	0.08	0.125
Suitability Factor	9.53	4.56	2.92	2.13	1.67
Positivity Factor	9	4	2.3	1.5	1

Let us dwell on the essence of entities appearing in Table 1.

The parameter α indicates the amount of chances that point realization of quality indicator will be less than the target mark, that is shows the magnitude of the risk on which the DM agree when sets a target mark. Clearly that the smaller the risk, which is accepted by DM, the smaller her/his target mark and vice versa. At the same time it is clear that this target mark can do not satisfy of DM but then she/he must accept a few larger risks.

Of course, in real cases the target mark must be positive. Positivity factor K_p shows at how many times, under the selected risk α, right boundary R of interval estimation must exceed its left boundary L (to be not less than this value) for the positivity of target mark. Naturally if DM may want to restrict himself by low risk then this factor may be too large and the analyzed interval can do not satisfy this requirement. DM must then either abandon the analysis or adopt more risk.

Suitability factor can be used in the analysis of interval estimates if DM desires to find those objects the risk factor values for which are less than the target mark.

One can see that this condition imposes stringent requirements on the right border of the interval estimation in comparison to the left so that the value of suitability factor exceeds the value of positivity factor. This difference is increased with growth of the risk, which DM accepts.

At last the value of the risk factor multiplied by the length of the analyzed interval object indicates the scope of the range where in the worst case may in average be point implementations of the quality indicator of the alternative. The lower the risk accepted by the decision maker the less is the scope of this range.

In concluding this section we distinguish the advantages and disadvantages of methods of "mean – risk" approach. An advantage of the methods of "mean – risk" approach is the possibility of calculating for each alternative both basic indicators needed for the evaluation interval alternatives, indicator of acceptability of alternatives and indicator of associated risk. The disadvantages include the fact that both of these indicators are calculated for an alternative as an independent object, which is not associated with the others in compared group. A comparison of the alternatives on preference is based on values of these indicators but dependence of the risk on the context is not taken into account. It should also be kept in mind that used here indicators are averaged estimates, which are not adequate, generally speaking, the situation of a unique choice.

At the same time in paper [Shepelev, 2014] was shown that results of interval objects selection on preference for the criterion of difference of mathematical expectations and the criterion of assurance factor are the same for a number of the simplest chances distributions. Recall that the assurance factor equals the difference in the chances of truth of tested hypothesis on preference of the analyzed alternative and in chances of truth of the opposite hypothesis.

Conclusion

Problems of comparing alternatives with numerical quality indicators, which due to uncertainty have interval representations, are fairly common in practice, in particular, in economics and engineering [Vilensky, 2015]. They play an important role and require special methods for their analysis. The inevitability of presence in similar tasks of irremovable risk of making a wrong decision on the preference of the alternatives requires human involvement in the decision-making processes. These processes are quite difficult; therefore DMs and experts need means of analytical support for their activity. They may permit to DMs and experts check how their knowledge and largely intuitive choice is consistent with the formal results and adjust their decisions. Besides use of different formal methods in the process of alternatives comparing and decision-making increases the volume and variety of accessible information that is useful to DMs and experts.

However formal methods of comparison and estimation of interval alternatives as a component of computer system for information-analytical support of the decision-making process cannot guarantee choice of truly the best object in the process of comparing or estimating. The results using of such methods can serve for DM only as a guideline, kind of a hint in the decision-making. At present there is no approach transcending all others in quality of recommendations obtained on its basis. Each method has its advantages and disadvantages.

Human involvement in the decision-making process can determine not only the choice of the tools of describing the uncertainty but also, due to previous experience and knowledge of human being, the choice of methods for comparison and estimation of alternatives that lead to the result indicators, which are familiar to the expert or DMs. Each of the available methods of comparison and evaluation allows calculating its measures of alternative preference in relation to the other alternatives in their set as well as indicators of the acceptability of alternatives during estimating and their risk measures.

Presence of the collective effect in groups of compared alternatives is manifested primarily in reducing value of preference chances for each alternative with respect to its chances under pair-wise comparison. This leads to a quantitative increasing risk value of selection as preferred alternative such one, which may not actually be per se later. The nature of this effect lies in the fact that in the presence of non-zero intersection for already two compared alternatives there is a non-zero risk of making the wrong decision. This risk is enhanced with increasing amounts of compared alternatives especially if some of these chances are not too different from each other. However, it should be borne in mind that the perception of risk is individual and can vary from one DM to another. Therefore the risk value resulting from the use of formal methods is nothing more than a calculated risk, which can serve only as an estimate for the DMs.

What can be done to reduce the calculated risk? Try to reduce the number of comparable alternatives as well as to take into account that joint using of different methods on various successive stages of the decision-making process is probably the best way to combine the power of formal methods and knowledge of experts and DMs.

Therefore decision-making process for the selection of interval alternatives is suggested in the paper to divide into three stages. Firstly, during deciding on preferred alternative choice or in the process of ordering alternatives by preference it’s useful to conduct a preliminary analysis of their initial set. After selecting an alternative that preference is tested it should select in the set of alternatives those, which do not have the intersection with analyzed alternative. If the left boundary of such intervals no less than the right boundary of the tested one the latter is certainly worse. If the right boundary of such intervals not greater than the left boundary of the tested one they can be excluded because they are certainly worse than the last interval. Then one may try to unify some similar or complementary alternatives. By reducing the number of intervals in their initial set one may increase the calculated preference chances of analyzed alternative and decrease collective risks.

Using of the method of collective risk estimating on the first stage of the decision-making process permits to appraise integral risks for each alternative in the group and find the "best" alternatives as the alternatives with the highest chances at pairwise comparisons in the set of compared alternatives.

At last, on the final stage, this narrowing set of alternatives should be evaluated according to the criteria of preference and risk, which are based on methods of "average – risk" approach. In the case of incomparability of alternatives to these criteria, when tested alternative is better by one criterion and worse by another, one may go to multi-criteria models or to models with one generalized optimality criterion. Both the initial criteria are harmonized in the generalized criterion by means of positive coefficient of replacement, which reflects the attitude of DM to take risks [Podinovsky, 2015].

One can expect that such a multi-step approach to decision-making on the choice of the preferred interval alternatives may contribute to increasing of adequacy of decision-making.

Acknowledgements

The paper is published with partial support by the project ITHEA XXI of the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua).

Bibliography

[Baker, 2015] Baker H.K., Filbeck G. Investment risk management. NY: Oxford university press. 2015.

[Fishburn,1977] Fishburn P.C. Mean-risk analysis with risk associated with below-target returns //American Economic Review. 1977. Vol. 67, pp.116 – 126.

[Halperin, 2010] Halperin B., Sevrin A. (Eds.). Quantum theory of condensed matter. World Scientific. 2010.

[Keeney, 1993] Keeney R.L., Raiffa H. Decisions with multiple objectives–preferences and value tradeoffs. Cambridge & New York: Cambridge University Press. 1993.

[Grechuk, 2012] Grechuk B., Molyboha A., Zabarankin M. Mean-deviation analysis in the theory of choice // Risk analysis. 2012. V. 32. P. 1277 – 1292.

[Larichev, 1979] Larichev O.I. Nauka i iskusstvo prinyatiya resheny. M.: Nauka. 1979. – in Russian

[Levy, 2006] Levy H. Stochastic dominance: investment decision making under uncertainty. New York: Springer. 2006.

[Nilsson, 1986] Nilsson N. J. Probabilistic logic. //Artificial Intelligence. 1986. Vol. 28, No. 1, pp. 71-87.

[Ogryczak, 1999] Ogryczak W, Ruszczyński A. From stochastic dominance to mean-risk models: semideviations as risk measures // European journal of operational research. 1999. V. 116, pp. 33 – 50.

[Petrovsky, 1996] Petrovsky A.B. Komputernaya podderzhka prinyatiya rezheny: sovremennoe sostoyanie i perspektivy // Sistemnye issledovaniya. Metodologicheskie problemy. Ezhegodnik 1996. M.: Editorial URSS. 1996 – in Russian.

[Podinovsky, 2015] Podinovsky V.V. Chislovye mery riska kak kriterii vybora pri veroyatnostnoi neopredelennosti //Iskusstvenny intellect i prinyatie resheny. 2015. V. 2, pp. 6 - 74 – in Russian

[Shepelyov, 2012] Shepelyov G., Sternin M. Modelling of preferences in comparing interval alternatives //Advances in Decision Technology and Intelligent Information Systems. 2012. V.13, pp. 31-35.

[Shepelyov, 2013] Shepelyov G., Sternin M. The applicability of mathematical expectation indicators in comparing interval alternatives. //Advances in Decision Technology and Intelligent Information Systems. Vol. 14, pp. 32-36. 2013.

[Shepelev, 2014] Shepelev G., Sternin M. The adequacy of point criteria during the evaluation and comparison of interval alternatives //Scientific and technical information processing. Vol. 41, No. 6, pp. 404 – 412. 2014.

[Shepelev, 2015] Shepelev G. Risk behaviour in a set of interval alternatives //International Journal "Information models and analyses". 2015. V. 4, pp. 303 – 323.

[Sternin, 2015] Sternin M., Shepelev G. Otsenka riska v sovokupnosti intervalnykh alternativ //Iskusstvenny intellect i prinyatie resheny. 2015. V. 3, pp. 83 - 91 – in Russian

[Vilensky, 2015] Vilensky P., Lifchits V., Smolyak S. Otsenka effektivnosti investitsionnykh proyektov. Moscow: PolyPrint Service. 2015.

Authors' Information

Gennady Shepelev – *Laboratory Head of Institute for Systems Analysis of Federal Research Center "Computer Science and Control" of RAS; e-mail: gis@isa.ru*

Major Fields of Scientific Research: *mathematical modeling, probabilistic methods, interval analysis, generalized interval estimations, comparing interval alternatives*

IMPLEMENTATION OF CONCURRENT CONTROL ALGORITHMS USING PLC LADDER DIAGRAMS

Liudmila Cheremisinova

***Abstract**: The problem of mapping a concurrent control algorithm onto a program structure for Programmable Logic Controller is discussed. The systematic method to derive Ladder Diagram programs from a parallel automaton that is a functional model of logic control device is presented. The mapping process is decomposed into a sequence of optimizing transformations of mathematical models of a parallel control algorithm specified in a formal language PRALU. The suggested procedures of optimizing the mathematical models are based on providing the proper control for a specific object under the control, i.e. within the restricted domain.*

***Keywords**: programmable logic controllers, parallel algorithms, control systems, state assignment.*

ACM Classification Keywords: *D. Software; D.1 PROGRAMMING TECHNIQUES; D.1.3 Concurrent Programming; D. Software; D.2 SOFTWARE ENGINEERING; D.2.2 Design Tools and Techniques.*

Introduction

Sequential control allows processing sequential and parallel operations in a discrete mode with respect to time or events. It is used to coordinate different continuous functions and to control complex process sequences. The behavior of the control system under discussion is characterized by complex interaction, asynchronism and concurrency. The widespread case is considered when a complex requires control in which inputs and outputs are on/off signals. The functions of a control of such a system are concentrated in one block – logic control unit that should provide proper synchronization of interaction between the components. In recent years the use of Petri net formalism has been gaining popularity for abstract description of the behavior of concurrent systems [Karatkevich, 2007, 2015].

The success of the control of a multiple component system greatly depends on the efficiency of the synchronization among its processing elements. The functions of a control of such a system are concentrated in a logic control device that should provide a proper synchronization of interaction between the components. In order to represent clearly the interaction involved in concurrent engineering system it is necessary to describe formally its functional and structural properties. This is becoming the usual industrial way to represent the control logic on the logical control level.

There exist many languages for description of logical control algorithms suited for these purposes. They can be divided into two classes: the languages formalizing methods of description of system functionality applied in industrial practice and languages based on formal mathematical models [Cheremisinova, 2002]. Keeping a logical control algorithm representation in mind a special language PRALU [Zakrevskij, 1989, 1999] has been chosen for these purposes. PRALU language combines the best features of languages from the mentioned two groups: it is clear enough for a designer and at the same time it has its background in Petri net theory (expanded nets of free choice – EFC–nets investigated by Hack [Hack, 1972]). The language has means for representation of asynchronous, sequential and concurrent processes. One of the proposed standard forms of PRALU–algorithms released from technical details was named as a parallel automaton [Zakrevskij, 1984]. This form is well suited for the purposes of hardware implementation of PRALU–algorithms and can be easily got from an initial PRALU–algorithm by its transforming [Zakrevskij, 1999].

The programmable logic controllers (PLC) [Bolton, 2015] are being used in many places where some kind of control is needed to run a real-time technical system. PLC-s [PLC, 1993] are digitally operating electronic systems designed for the use in an industrial environment and they are now widely used in many technical “real world” applications such as complex petro-chemical plants, robotic centers, automobile production lines.

The problem of design of logic control devices for distributed discrete-event systems is considered. The widespread case is considered when a technical system requires control in which inputs and outputs are on/off signals. The problem of mapping a parallel control algorithm onto a program structure for PLC is discussed. We use the most popular way of PLC programming based on the use of Ladder Diagram (LD) language. The functional mathematical model of LD program is suggested and it is shown that at the heart of the suggested method the asynchronous hardware realization of parallel automaton lies. The paper presents a systematic method to derive LD programs from a parallel automaton that is a functional model of logic control device. The mapping process is decomposed into a sequence of optimizing transformations of mathematical models of a parallel control algorithm. The proposed optimization procedures are based on providing the proper control for a specific object under the control. This possibility for optimization arises when considering the control unit behavior together with the behavior of the controlled object.

Programmable Logic Controllers

PLC is a special form of microprocessor-based controller that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes. PLC-s were invented to replace the sequential electromagnetic relay circuits. They work by looking at their inputs and depending upon their state, switching on/off their outputs. To get the desired behavior of a control device a designer has to write a program on some of the available languages that is then translated into a list of instructions that have one-to-one correspondence to each symbol of the language.

Many programming languages have been developed for programming PLCs. Each manufacturer has its own language. To improve the software used in PLCs, the International Electrotechnical Commission (IEC) has defined a standard for programming languages which is called IEC 1131-3 [PLC, 1993]. The IEC 1131-3 standard tries to bring together the languages from the PLC and the computing worlds. The following is a list of programming languages specified by this standard:

1) Ladder Diagram (LD) that has been developed to mimic relay logic;

2) Sequential Function Charts (SFC) that is similar (but much more powerful) to flowcharts, it was developed to accommodate the programming of more advanced systems;

3) Function Block Diagram (FBD) that represent PLC programs as connection of different function blocks;

4) Structured Text (ST) that is a textual (PASCAL or BASIC like) programming language;

5) Instruction List (IL) that is an assembly like language;

The first three languages are graphical ones and the last two are textual languages.

A very commonly used method of programming PLCs is based on the use of ladder diagrams [Bolton, 2015]. Writing a program is then equivalent to drawing a switching circuit. Each LD diagram is composed of two vertical lines representing the power rails. The power rails are connected as horizontal lines called as rungs of the ladder, between these two verticals. In drawing a ladder diagram, certain conventions are adopted:

1) The vertical lines of the diagram represent the power rails between which circuits are connected. The power flow is taken to be from the left-hand vertical across a rung.

2) Each rung on the ladder defines one operation in the control process.

3) A ladder diagram is read from left to right and from top to bottom.

4) When the PLC is in its run mode, the rungs of an LD program are executed until an end rung is reached. The procedure of going through all the rungs of the program is termed a cycle.

5) Each rung must start with one or more inputs and end with at least one output.

6) A particular device can appear in more than one rung of a ladder. A relay may switch on one or more devices.

PLC language instructions can be classified into two categories: logic instructions and block instructions. Practically all PLCs have the same set of logic instructions having one-to-one correspondence with coils and contacts of the relay circuit. Accordingly to that the program variables can be divided into logic (Boolean) (including input and output ones) and arithmetic variables. As we aimed to consider the control part of a technical system the set of logic instructions is of the main interest (the other operations are translated to logical ones when transforming a control algorithm into a parallel automaton [Cheremisinova, 1988]. Practically all PLCs has the same set of logic instructions, the typical set of logic instructions includes “examine if on (or off)”, “set on (or off)”, “set on (or off) and preserve”. Table 1 shows the typical set of basic logic instructions (as in [Allen-Breadly, 1976]). An example of a LD-program rung is shown in Figure 1.

In the most basic form a PLC is a microprocessor-based controller that executes an application program by interpreting its instructions. When executing an program a PLC continuously repeats a single loop called a cycle which consist of 3 steps: sampling input signals, executing an program to update the PLC’s internal registers and delivering new output signal.

We mentioned before that an LD program is a list of rungs. From a syntactic point of view we can decompose a rung into two parts: a front and a rear. A front corresponds to a logic formula over the rung inputs. An input can be normally open or close. The sequential and parallel connections of the inputs represent respectively conjunction and disjunction. So, the left part of the logic formula is a multilevel Boolean expression over AND, OR, NOT operations. A rear of a rung is one or more outputs connected in parallel. In a rear, an output can be preceded by a front circuit like in the example of Figure 1.

Table 1. Basic logic instructions of LD language

<table>
<tr><th>Name</th><th>Graphic symbol</th><th>Operation</th><th>Result of instruction: is TRUE if</th><th>Relay analog</th></tr>
<tr><td>Load</td><td>–| |–</td><td>Examine if on</td><td>Input is TRUE</td><td>Normally open contact</td></tr>
<tr><td>LoadNot</td><td>–|/|–</td><td>Examine if off</td><td>Input is FALSE</td><td>Normally closed contact</td></tr>
<tr><td>Out</td><td>–( )–</td><td>Set on</td><td>There is a path of TRUE instructions on the rung</td><td>Relay coil</td></tr>
<tr><td>OutBar</td><td>–(/)–</td><td>Set off</td><td>There is a path of FALSE instructions on the rung</td><td>Doesn't exists</td></tr>
<tr><td>Set</td><td>–(L)–</td><td>Set on and preserve</td><td>There is a path of TRUE instructions on the rung</td><td rowspan="2">Latching relay</td></tr>
<tr><td>Reset</td><td>–(U)–</td><td>Set off and preserve</td><td>There is a path of TRUE instructions on the rung</td></tr>
</table>

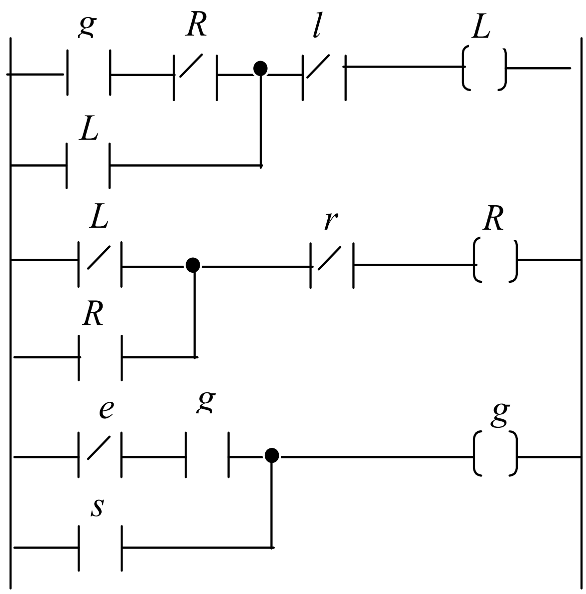


Figure 1. An example of LD-program

Ideally all rungs or as well as Boolean equations are independent and thus they should be evaluated simultaneously but in practice rungs are evaluated in a top-to-bottom order as they appear in a program. LD rung as well as a Boolean equation may be regarded as an if-then statement in programming languages. If-clauses are composed of logic instructions or compare instructions. Then-clauses are composed of output or arithmetic instructions. When a condition of an if-clause becomes true the corresponding then-clause is executed.

To answer the question what are the values of variables, defined by Boolean expressions represented by a system of Boolean equations or LD-program we have to think about the scanning sequence. The first thing in the scan cycle is to read the values of input signal from PLC sensors into the memory. Next, the PLC executes a program starting from the top left to bottom, changing the values of Boolean variables and using these new values when executing then-clauses following just after their setting.

Parallel and sequent automata

The process of a control algorithm mapping into PLC program can be understood as a sequence of transformations of mathematical models of the control PRALU-algorithm. When solving the problem of a control algorithm implementation one faces the necessity of getting first a formal finite-state model of the algorithm called as a parallel automaton [Zakrevskij, 1984]. This automaton model has structural input and output states and abstract internal ones. An essential difference of parallel automaton model from sequential one is that it can be in more than one partial state simultaneously. In that case the partial states are called parallel. All parallel partial states, a parallel automaton is in at some moment, form its global state. Any transition of automaton defines changes of partial states that cause the current global state change. As well as a customary sequential automaton parallel automaton may be of Moore type or Mealy type.

For the purposes of PLC implementation of a control algorithm it is useful (though more difficult) to get a Moore type automaton. It is obtained by cutting long chains of the PRALU-algorithm on a set of strings (interpreted as automaton transitions) in the form:

$$\mu_n : -k_n^{in} \to \nu_n / k_n^{out},$$

where k_n^{in} and k_n^{out} are elementary conjunctions of Boolean variables, μ_n and ν_n are initial and terminal labels: subsets of natural numbers – states. The expression “$-k_n^{in}$” denotes the waiting operation of the PRALU-language: to wait until the term k_n^{in} takes the value 1. “$\to k_n^{out}$” denotes the acting operation: to give such values to the variables from conjunction k_n^{out} it to be equal to 1. k_n^{in} and k_n^{out} are interpreted as the input condition of the transition and the output signals respectively, μ_n and ν_n are interpreted as subsets of partial states s_i. Such a transition should be understood as follows: if the current global state of the parallel automaton contains all the partial states from μ_n and the variables in the conjunction term k_n^{in} assume values providing k_n^{in} = 1, then (as the result of the transition) the automaton goes to the next global state involving from initial one by substituting partial states from μ_n for the states from ν_n.

The values of inner and output variables of Moore type automaton are specified by partial states. Note that the conjunction k_n^{out} is divided into $|\nu_n|$ parts k_{ni}^{out}: each $k_{ni}^{out} \in k_n^{out}$ corresponds to some partial state $s_i \in \nu_n$. Taking into account the most used behavioral interpretation of output variables of initial PRALU-algorithm it can be said that automaton considered is inertial [Zakrevskij, 1999] over the set of its output variables, i.e. the variables that are absent in k_n^{out} preserve their values.

For instance, let us consider the process of transforming the following PRALU-algorithm (from [Zakrevskij, 1999]):

$$1: -u \to ab - \bar{u} \to 2.3$$

$$2: -\bar{v}w \to \bar{b}c - \bar{w} \to b\,\bar{c} \to 2$$

$$-v \to \bar{a}c \to 4.5$$

$$3: -uw \to d \to 6$$

$$4: -u \to a\,\bar{b} \to 7$$

$$5: -\bar{v}w \to \bar{c} \to 8$$

$$6.7.8: \bar{u}v \to \bar{a}\,\bar{d} \to 1$$

into a parallel automaton of Moore type:

$$1{:}\ |_1 - u\ |_9 \to a\,b - \bar{u} \to 2.3$$

$$2{:}\ |_2 - \bar{v}\,w\ |_{10} \to \bar{b}\,c - \bar{w}\ |_{11} \to b\ \bar{c} \to 2$$

$$- v\ |_{12} \to \bar{a}\,c \to 4.5$$

$$3{:}\ |_3 - u\,w\ |_{13} \to d \to 6$$

$$4{:}\ |_4 - u\ |_{14} \to a\ \bar{b} \to 7$$

$$5{:}\ |_5 - \bar{v}\,w\ |_{15} \to \bar{c} \to 8$$

$$6.7.8{:}\ \bar{u}\,v\ |_{16} \to \bar{a}\ \bar{d} \to 1$$

After simplifying the automaton and renumbering its partial states [Cheremisinova, 2002] we have:

$$u \quad s_1 \to s_9\ /a\ b$$

$$\bar{u} \quad s_9 \to s_2 \cdot /b\ \bar{c}.\ s_3\ /\text{-}$$

$$\bar{v}\,w \quad s_2 \to s_{10}\ /\ \bar{b}\,c$$

$$\bar{w} \quad s_{10} \to s_2\ /b\ \bar{c}$$

$$v \quad s_2 \to s_4\ /\ \bar{a}.\ s_5\ /c$$

$$u\,w \quad s_3 \to s_6\ /d$$

$$u \quad s_4 \to s_7\ /a\ \bar{b}$$

$$\bar{v}\,w \quad s_5 \to s_8\ /\ \bar{c}$$

$$\bar{u}\,v \quad s_6.s_7.s_8 \to s_1\ /\ \bar{a}\ \bar{d}$$

Traditionally, the next step on the way to control device hardware implementation is state assignment. A peculiarity of this process for parallel automaton is that there are parallel states in it. It was suggested [Zakrevskij, 1989] to code partial states with ternary vectors which should be non-orthogonal for parallel partial states (but orthogonal for non-parallel). After encoding partial states an initial parallel algorithm can be transformed from its abstract form into a structural one – sequent automaton that can be directly hardware implemented.

A sequent automaton is a system of sequents [Zakrevskij, 2000] $f_i(V)$ |– k_i, where $f_i(V)$ is a Boolean function over variables from $V = X \cup Y \cup Z$ and k_i is some elementary conjunction of variables from $W = Y \cup Z$, where X, Y and Z are sets of input, output and inner variables. Such an expression is specified semantically by the following manner: once the Boolean function $f_i(V)$ equals 1 then immediately after that the variables from k_i should be set to be such that k_i equals 1. A set of values of all variables from V defines a total variable vector-state $\boldsymbol{v}$ of the sequent automaton. Below a sequent $f_i(V)$ |– k_i is called active on the state $\boldsymbol{v}^k$ if $f_i(\boldsymbol{v}^k)$ is true.

Further sequent automata inertial in relation to a subset $U \subseteq V$ will be considered. A sequent $f_i(V)$ |– k_i of such an automaton defines the following relationship on the set U: if the current state $\boldsymbol{v}^k$ of sequent automaton is such that some sequent is active on it then in the next state $\boldsymbol{v}^{k+1}$ all variables from U except for those mentioned in the conjunction k_i keep their values. Thus the variables from U keep their values, that have got, until any active sequent changes them. This property implies the interpretation of acting operations of the PRALU-language and forces to implement all output variables with flip-flops as well as inner variables.

Two types of sequent automata will be dealt with: a simple and a functional sequent automata [Zakrevskij, 1999]. The first one is a system of simple sequents k_i' |– k_i''. The functional sequent automaton consists of the sequents having the form $f_i^1(V)$ |– w_i or $f_i^0(V)$ |– $\overline{w}_i$, where $w_i \in W$ and $f_i^1(f_i^0)$ are in the disjunctive normal form. In fact such an automaton is a system of Boolean equations. Having in view a Moore type automaton any its transition $\mu_n : -k_n^{in} \rightarrow \nu_n / k_n^{out}$ will be converted into 1 + $|\nu_n|$ simple sequents, i.e. it generates the following sequents:

$$k_n^{in'} \wedge k^{\mu} \,|\!-\, k^{\nu}, \quad k^{\nu}_i \,|\!-\, k_{ni}^{out}, \; i = 1, 2,\ldots, |\nu_n|,$$

where k^{μ} and k^{ν} are conjunctive terms defining unions of compatible codes of partial states from the sets μ_n and ν_n respectively, k^{ν}_i and k_{ni}^{out} are conjunctive terms defining the code of the partial state $s_i \in \nu_n$ and output signals k_n^{out} implied by s_i.

Deriving LD realizable Boolean equations

As we focus our discussion on design of a control device let us consider a control part of the LD program. Omitting technical details it can be said that the control part of LD program is an ordered set of logic equations $F_i(V) = w^{\sigma}_j$, where $w_j^{\sigma} \in W = Y \cup Z$ ($w_j^{\sigma} = w_j$ when σ = 1 or $\overline{w}_j$ when σ = 0). $F_i(V)$ defines multilevel Boolean expression over AND, OR, NOT operations. Omitting brackets in the expressions $F_i(V)$ converts them into a sum of products form (disjunctive normal form). Thus a LD program can be represented syntactically by a functional sequent automaton.

When PLC executes a program the discipline of changing internal states should be maintained by input signals. In other words, there must be a stable state on W for any input state on X. In a similar, an asynchronous sequent automaton finding itself in a stable state $\boldsymbol{w}^k$ passes into another stable state $\boldsymbol{w}^{k+1}$ after changing an input state $\boldsymbol{x}^k$. A transition between stable states could be fulfilled as a result of execution of one or more active sequents. The execution order of sequents defines a variant of a sequence of intermediate states $\boldsymbol{w}_i^{k+1}$ that take place when leaving $\boldsymbol{w}^k$ for $\boldsymbol{w}^{k+1}$. This order is random for asynchronous automata.

As to the LD program, Boolean equations (or LD-rungs) are executed in turn and the result of their execution is used in the next equations. Strictly speaking, the derived sequence of intermediate states is not random – one can find it for any given $\boldsymbol{w}^k$ and $\boldsymbol{x}^k$. But to provide an automaton to function properly the assumption, that the sequence of changes of inner variable values (as the result of input state modification) is purely arbitrary, is sufficient for proper modeling control algorithm by LD program. Thus the first and the basic step that should be done to convert parallel automaton into LD program is to synthesize an asynchronous inertial functional sequent automaton.

It is possible to synchronize changes of inner variables values during scanning the LD program by means of reduplication of inner variables. That is, one more inner variable z_i' is introduced for every variable $z_i \in Z$. Then z_i is used only in the left parts of Boolean equations, so it preserves invariably its value during the whole cycle of LD program scanning in top-to-bottom. z_i' is used only in the right parts of Boolean equations, so it can get new value in accordance with LD program flow. At the end of the LD program cycle z_i is set to have a value of z_i' and these new values of inner variables are used to set values of output variables. Such synchronization redoubles the number of inner variables but makes it possible to use state assignment methods developed for automaton synchronous implementation.

State assignment of asynchronous parallel automaton

When automaton asynchronous implementation is considered the additional condition has to be fulfilled to avoid the influence of races between memory elements during hardware operation. One of the ways to avoid them is to order memory elements switches so as to eliminate critical races [Cheremisinova, 2004]. The drawback of this approach seems to be the "density" of the codes obtained.

By using PLC for control device implementation it seems to be convenient to use a trivial method of encoding the partial states – 1-hot encoding. For example, such a coding had been used for PLC implementation of Graphset [Chevalier, 1980]. In this case the number of encoding variables is equal to the number of internal partial states of an automaton. The code of any partial state of a parallel automaton will have the only unit component. Any other component will be 0 if it corresponds to a partial state incompatible with considered one or don't care ("-") if it corresponds to a partial state parallel to considered one. But for some moment it might be more than one memory elements set to 1 due to existence of parallel partial states.

Considering a transition between two partial states it should be noted that it passes through one intermediate state encoded by the code with two unit components. This unstable state is alternated with the force of the same input signal to resulting one. In general case a transition from n partial states to m ones takes place over $n + m$ -1 unstable states.

Having in mind an asynchronous implementation of parallel automaton it is desirable to agree on which of two output signals will be displayed during unstable states taking place in the transition $\mu_n \to \nu_n$: one implied by μ_n or the other one implied by ν_n. For distinctness let us agree that the output signals begin to change over their values in the state imposing ν_n immediately after switching on encoding variables of all partial states from ν_n. Then only those states can cause the output variable settings that are at the end of a LD program scan. So, if there exists an unstable state that takes place and that is changed within a scan cycle it cannot do necessary output variable setting. A partial state s_i is an unstable if there exist the following two transitions in the automaton:

$$\mu_n : -k_n^{in} \to \nu_n / k_n^{out}, \ \mu_p : -k_p^{in} \to \nu_p / k_p^{out} \ (s_i \in \nu_n, s_i \in \mu_p)$$

having compatible (nonorthogonal) input conditions k_n^{in} and k_p^{in}. If there exist such an unstable state the appropriate encoding variable should be doubled when implementing the automaton with LD program. Otherwise it is not necessary.

Sequent automaton mapping into LD program

Let us suppose that each partial state s_i of an automaton is encoded by inner variable z_i. And let the global state and the input state $\boldsymbol{x}_t$ of the automaton are such that the transition $t_n = \mu_n : -k_n^{in} \to \nu_n / k_n^{out}$ takes place. When converting a parallel automaton to functional sequent one it should be paid attention to following.

1. The inner variable z_i ($s_i \in \nu_n$) is switched on immediately after the transition t_n becomes active.

2. The inner variable z_j ($s_j \in \mu_n$) is switched off only after all inner variables z_i ($s_i \in \nu_n$) take the values 1.

3. The output variable y_p preserves its value implied by $s_i \in \mu_n$ until the automaton reaches at least one of the partial states from ν_n controlling y_p.

4. The output variable y_q switches the proper value implied by $s_j \in \nu_n$ immediately after reaching s_j.

For any variable $w_p \in W$ we form two functions f_p^1 and f_p^0, which present in fact its on- and off-functions. Let us choose for every partial state s_p sets T_p^{in} and T_p^{from} of all transitions into and from it respectively: $t_i = \mu_i : -k_i^{in} \to \nu_i / k_i^{out} \in T_p^{in}$ if $s_p \in \nu_i$ and $t_i \in T_p^{from}$ if $s_p \in \mu_i$. Then in accordance with what was said the functions f_p^1 and f_p^0 have the following general forms:

$$f_p^1 = \bigvee_{t_j \in T_p^{in}} (k_j^{in} \bigwedge_{s_i \in \mu_j} z_i) , \quad f_p^0 = \bigvee_{t_j \in T_p^{from}} (\bigwedge_{s_i \in \nu_j} z_i) .$$

As to the output variables let us pick up all partial states implying the values 1 and 0 of the variable $y_q \in Y$ into the sets S_q^{on} and S_q^{off} respectively. Then the functions f_q^1 and f_q^0 will have the following forms:

$$f_q^1 = \bigvee_{s_i \in S_q^{on}} (z_i \overline{\bigvee_{s_r \in S^{iq}} z_r}) , \quad f_q^0 = \bigvee_{s_i \in S_q^{off}} (z_i \overline{\bigvee_{s_r \in S^{iq}} z_r}) .$$

Here S^{iq} depends on s_i and y_q: $S^{iq} = \{s_r / (y_q \in C(s_r))$ & $(s_r \in \nu_j)$ & $(t_j \in T_i^{from})$, where $C(s_r)$ denotes a set of variables controlled by the partial state s_r [Cheremisinova, 1988].

Minimization of the code length

The shortcoming of 1-hot state encoding is that the number of introduced inner variables is overmuch. But it seems to imply a fast program having rare rungs (or equations). At the same time it should be taken into account that the number of encoding variables can be considerably decreased (sometimes to 0) at the expense of using the values of some existing variables of an automaton as encoding ones. They can separate some incompatible partial states.

The set X of input variables of the control device can be divided into two subsets X_1 and X_2: those arriving from an outside environment (from human operator, for example) and those from the object under the control. The variables from X_1 can be considered as the inputs of the control system as a whole (with the controlled object) and, generally speaking, their behavior is unknown. The variables from X_2 are the internal variables for the control system. Their values depend on the object under the control. How the values of these variables will change, one can predict if the behaviour of the object under the engineering system control is known. PRALU-language allows to describe together the behavior algorithms of the object and the control device [Cheremisinova, 1989]. In such a description of the whole control system, variables from X_2 can be considered as inertial ones. Besides variables from X_2 output variables can be added to the set U of inertial variables. Just inertial variables are allowed to use as encoding for partial state assignment. That is permissible if their values are known in any automaton state controlling these variables. The conjunction of the values, which these variables have, when the automaton is in a partial state, can be used as a part of its code. For these purposes the conjunction terms k_n^{in} and k_n^{out} of a parallel automaton transitions $\mu_n : -k_n^{in} \to \nu_n / k_n^{out}$ are expanded up to the conjunctions k_n^{in*} and $k_n^{\nu*}$ of the generated sequent $k_n^{in*} \wedge k_n^{\mu} |- k_n^{\nu*}$. The method of calculating a set of all possible states on the set U includes defining [Cheremisinova, 1988*]:

1) concurrency relation on the set of partial states;

2) sets (non-overlapping for parallel partial states) of variables from *U* controlled by partial states;

3) set of eventual values of the variables from *U* that are controlled by every given partial state.

The first problem is defined [Cheremisinova, 2004] on a dynamic model of a parallel automaton (or a control algorithm) – its “skeleton” called as α-net [Zakrevskij, 1999]. When the concurrency relation has been found, for any partial state it can be said what partial states it is parallel to. A variable is called as “controlled” by a partial state s_k if its value can be changed when automaton finds itself in s_k and it may be changed in no partial state parallel to s_k. The subset of all such variables for a partial state s_k includes all variables under consideration except those changing their values in partial states parallel to s_k. For the considered above parallel automaton partial states s_2, s_4, s_5, s_7 and s_8 are parallel to s_3 and s_6; s_4 and s_7 – to s_5 and s_8. s_1 control all output variables {*a*, *b*, *c*, *d*}; s_3 and s_6 – the only *d*; s_4 and s_7 – {*a*, *b*}; s_5 and s_8 – the only *c*.

Total variable states are considered on a set *U* of variables the automaton is inertial in relation to it. Each possible global state $\boldsymbol{s_k}^*$ defines a set S_k^* of partial states and appropriate total variable state $\boldsymbol{u}_k^*$ on the set of variables controlled by partial states from S_k^*. This state $\boldsymbol{u}_k^*$ consist of non-orthogonal (by definition) variable states $\boldsymbol{u}_i$ corresponding to partial states $s_i \in S_k^*$. In [Cheremisinova, 1988] a concept of initial and final variable states of transitions of parallel automaton was introduced. An initial variable state $\boldsymbol{u}_n^b$ of a transition $t_n = \mu_n : -k_n^{in} \rightarrow \nu_n / k_n^{out}$ is a vector of values that might have the variables controlled by partial states $s_i \in \mu_n$ when that transition takes place. The final variable state $\boldsymbol{u}_n^e$ of the transition differs from appropriate initial one in values of variables from k_n^{out}.

An initial variable state $\boldsymbol{u}_n^b$ is a concatenation of component-vectors $\boldsymbol{u}_{nj}^b$ of values, variables controlled by partial states $s_j \in \mu_n$ might have. $\boldsymbol{u}_n^b$ implies each of $\boldsymbol{u}_{nj}^b$ (they are compatible). Each vector $\boldsymbol{u}_n^b$ component $\boldsymbol{u}_{nj}^b$ is defined by final variable states of transitions t_i: $\mu_i \rightarrow \nu_i$ to the partial state $s_{nj} \in \mu_i$ and $s_{nj} \in \nu_i$. The set of initial and final variable states of all automaton transitions gives the entire domain of the control system definition. If the variable state $\boldsymbol{u}_j$ implicating component-vectors $\boldsymbol{u}_{nj}^b$ of all initial variable states of all transitions t_n: $\mu_n \rightarrow \nu_n$, $s_j \in \mu_n$ is found, it may be considered as a part of the code of s_j on the set of variables controlled by it.

Let us consider the example of the reciprocating motion from [Allen-Breadly, 1976] (Figure 2). The object is to start out at the right hand of the table, activating the limit switch *r*. When the “start” pushbutton (the appropriate variable *s*) is pressed, right to left motion should occur until the limit switch *l* is tripped. Then left to right motion occurs until the limit switch *r* is tripped. This cycle should continue until the limit switch *r* is tripped again after the “stop” (the appropriate variable *e*) pushbutton is pressed. At this point motion will stop until “start” is pressed again. The output signals *L* and *R* of the control device are introduced to cause right to left and left to right motions correspondingly, as shown in Figure 2.

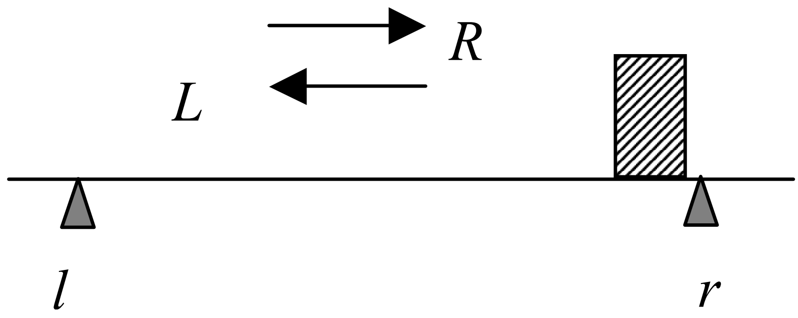


Figure 2. Example of reciprocating motion.

The following PRALU-algorithm describes the behavior of the control needed to provide such a reciprocating motion (the internal variable *g* is introduced to indicate the last of the pressings "start" or "stop"):

Motion(s, e, l, r / L, R)

1: $\rightarrow 2.3$

2: $- g \rightarrow L - l \rightarrow \bar{L} \rightarrow R - r \rightarrow \bar{R} \rightarrow 2$

3: $- s \rightarrow g - e \rightarrow \bar{g} \rightarrow 3$

The following PRALU-algorithm describes the behavior of the whole control system:

Motion(s, e, l, r / L, R)

$\bar{l}\, r\ \bar{g}\ \bar{L}\ \bar{R}$

1: $\rightarrow 2.3$

2: $- g\ \bar{R}|_4 \rightarrow L \rightarrow \boldsymbol{\bar{r} \rightarrow l} - l\,|_5 \rightarrow \bar{L} - \bar{L}|_6 \rightarrow R \rightarrow \boldsymbol{\bar{l} \rightarrow r} - r\,|_2 \rightarrow \bar{R} \rightarrow 2$

3: $- s\,|_7 \rightarrow g - e\,|_3 \rightarrow \bar{g} \rightarrow 3$

Here $X = \{s, e, l, r\}$, $Y = \{L, R\}$, $Z = \{g\}$ are the sets of input, output and internal variables. $\bar{l}\, r\ \bar{g}\ \bar{L}\ \bar{R}$ specifies the values of the variables, the automaton is inertial to, immediately before the motion starting. The part of PRALU-algorithm describing the behavior of the object of the control is set off in bold face. The waiting operations "– $\bar{L}$" and "– $\bar{R}$" (not changing the control algorithm) have been introduced to do the control algorithm as LD program realizable without doubling inner variables. The variables l and r define the behavior of the object of control and they are internal for a control system as a whole, so their exhaustive behavior can be defined as well as for internal and output variables of the control device: g, L and R. Thus the variables from $U = \{l, r, g, L, R\}$ may be used as encoding ones for automaton partial states.

The first column of Table 2 shows the obtained parallel automaton of Moore type realizing the behavior of the control system. The sets of initial and final variable states of transitions of the automaton on the set U are given having in mind that the partial states from $\{s_2, s_4 – s_6\}$ and $\{s_3, s_7\}$ control the variables from $U_2 = \{l, r, L, R\}$ and $U_3 = \{g\}$ (the values of uncontrolled variables are marked as “+”).

Table 2. Parallel automaton of Moore type

The automaton transitions	Initial states $l\ r\ g\ L\ R$	Final states $l\ r\ g\ L\ R$
t_1: s_2:– g – $\overline{R} \to s_4 \to L$	0 1 $\underline{0}$ 0 $\underline{0}$	0 1 + 1 0
t_2: $\to \overline{r}$	**0 1 + 1 0**	**0 0 + 1 0**
t_3: $\to l$	**0 0 + 1 0**	**1 0 + 1 0**
t_4: s_4:– $l \to s_5 \to \overline{L}$	$\underline{1}$ 0 + 1 0	1 0 + 0 0
t_5: s_5:– $\overline{L} \to s_6 \to R$	1 0 + $\underline{0}$ 0	1 0 + 0 1
t_6: $\to \overline{l}$	**1 0 + 0 1**	**0 0 + 0 1**
t_7: $\to r$	**0 0 + 0 1**	**0 1 + 0 1**
t_8: s_6:– $r \to s_2 \to \overline{R}$	0 $\underline{1}$ + 0 1	0 1 + 0 0
t_9: s_3:– $s \to s_7 \to g$	+ + 0 + +	+ + 1 + +
t_{10}: s_7:– $e \to s_3 \to \overline{g}$	+ + 1 + +	+ + 0 + +

The partial states s_2, s_4, s_5 and s_6 (as well as s_3 and s_7) of the control automaton are pair-wise incompatible and their codes should be orthogonal. But we see that their initial states are parewise orthogonal with respect to variables from $\{l, r, L, R\}$ ($\{g\}$). So we need no more additional variables to encode the automaton partial states. So, the following functional sequent automaton realizing realizes the parallel automaton (shown in the first column of Table 2):

$$\bar{l}\,r\,g\,\bar{L}\,\bar{R} \mid\!- L$$

$$l\,\bar{r}\,L\,\bar{R} \mid\!- \bar{L}$$

$$l\,\bar{r}\,\bar{L}\,\bar{R} \mid\!- R$$

$$\bar{l}\,r\,\bar{L}\,R \mid\!- \bar{R}$$

$$s\,\bar{g} \mid\!- g$$

$$e\,g \mid\!- \bar{g}$$

As follows from the parallel automaton description (Table 2) there exist no unstable partial states. So implementing the sequent automaton with LD program there is no need to duplicate the variables *L* and *R* that are used as inner ones. Then taking into account the sequential mode of LD execution we can simplify the sequent automaton by discarding some inner variables from sequents:

$$\bar{l}\,g\,\bar{R} \mid\!- L$$

$$l \mid\!- \bar{L}$$

$$\bar{r}\,\bar{L} \mid\!- R$$

$$r \mid\!- \bar{R}$$

$$s \mid\!- g$$

$$e \mid\!- \bar{g}$$

As one could see from Table 1 any Boolean function can be realized by means of two LD instructions: a simple relay coil or latching relay (latch-unlatch). In the first case two sequents for the same variable (of a functional sequent automaton) is realized as a single rung of LD program, in the second case – as two rungs.

Taking into account that the sequent automaton is inertial one over inner and output variables we can realize any pair of sequents defining w_i and $\bar{w}_i$ by means of one of two LD instructions. That is Set-Reset (latching relay) instruction) or more preferable Out instruction (its relay coil analog). So, any pair of sequents $w_i = f_i^1$ and $w_i = f_i^0$ could be implemented as:

$$w_i = f_i^1 \vee w_i\ \overline{f_i^0} \text{ or } w_i = (f_i^1 \vee w_i)\ \overline{f_i^0}$$

The appropriate LD program is shown in Figure 1.

Conclusion

This paper presents the process of mapping control algorithm onto PLC program as a sequence of transformations of mathematical models of PRALU-algorithm. The task of minimization of PLC program (in the form of LD diagram) length is achieved as a result of solution of some optimization problems concerning transformations of PRALU-algorithm models.

Acknowledgement

The paper is published with partial support by the project ITHEA XXI of the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua).

Bibliography

[Karatkevich, 2007] A.Karatkevich. Dynamic Analysis of Petri Net-Based Discrete Systems. In: Lecture Notes in Control and Information Sciences. Springer -Verlag, vol. 356, 2007, 166 p.

[Karatkevich, 2015] A.Karatkevich. Petri Nets in Design of Control Algorithms. In Design of Reconfigurable Logic Controllers. Springer -Verlag, Vol. 45 of the series Studies in Systems, Decision and Control, 2015, pp 1-14.

[Zakrevskij, 1989] A.D.Zakrevskij. To the theory of parallel algorithms for logical control. In Izvestiya AN SSSR. Tekhnicheskaya Kibernetika, 1989, No. 5, pp. 179–191 (in Russian).

[Zakrevskij, 1999] A.D.Zakrevskij. Parallel algorithms for logical control. Minsk: Institute of Engineering Cybernetics of NAS of Belarus, 1999. (in Russian).

[Hack, 1972] N.Hack. Anaysis of production schemata by Petri nets. Project Project MAC TR-94, Cambridge, 1972.

[Zakrevskij, 1984] A.D.Zakrevskij. Parallel automaton. In Doklady AN BSSR, 1984, vol. 28, No. 8, pp. 717 – 719 (in Russian).

[Bolton, 2015] W.Bolton. Programmable Logic Controllers, Elsevier, 2015, 6th Edition, 424 p.

[PLC, 1993] International Electrotechnical Commission. Programmable Controllers. Part 3. Programming Languages, IEC Publication, 1993, pp. 1131-1133.

[Allen-Breadly, 1976] Allen-Breadly Company. PLC Programming and operations manual, 1976, Bulletin 177.

[Cheremisinova, 2002] L.D.Cheremisinova. Implementation of parallel algorithms of logical control. Minsk: Institute of technical Cybernetics of NAS of Belarus, 2002, 246 p.

[Zakrevskij, 2000] A.Zakrevskij, B.Steinbach. Sequent automaton - a model for logical control. In Proceedings of the Intern. Workshop "Discrete optimization methods in scheduling and computer-aided design", Minsk: Republic of Belarus, Sept. 5–6, 2000.

[Cheremisinova, 2004] L.D.Cheremisinova. State Assignment of an Asynchronous Parallel Automaton. In Journal of Computer and Systems Sciences International, 2004, vol. 43, No. 5, p. 743–750.

[Chevalier, 1980] G.Chevalier. Le Grafcet: les automatismes par le diagramme fonctionnel et la technologie modulaire. Paris: Dunod, 1980.

[Cheremisinova, 1988] L.D.Cheremisinova, V.K.Vacilenock, E.V.Sheludko, L.V.Krasilnikova. The system of logical design of control devices on the base of programmable controllers. Minsk: Institute of Engineering Cybernetics of the of Belarus Academy of Sciences, 1988, 100 p. (in Russian).

[Cheremisinova, 1989] L.D.Cheremisinova. The methods of automaton realization of concurrent control algorithms expressed on PRALU language. Minsk: Institute of Engineering Cybernetics of the of Belarus Academy of Sciences, 1989, 58 p. (in Russian)

[Cheremisinova, 1988*] L.D.Cheremisinova. Minimization of finite-response sequential automata that implement parallel logical control algorithms. In Automatic Control and Computer Sciences, 1988, vol. 22, No. 4, pp. 69 – 74.

[Cheremisinova, 2004] L.D.Cheremisinova. Concurrency relation and automaton realization of logical control algorithms. In Proceedings of the Fifth Intern. Conf. on Computer-Aided Design of Discrete Devices, CAD DD'2004, v.1, Minsk: Rep. of Belarus, Nov.16–17, 2004, pp. 112 – 120.

Authors' Information

Liudmila Cheremisinova – *The United Institute of Informatics Problems of National Academy of Sciences of Belarus, principal recearcher, Surganov str., 6, Minsk, 220012, Belarus; e-mail: cld@newman.bas-net.by*

Major Fields of Scientific Research: *Discrete mathematics, Logic design automation*

MULTI-AGENT DESCRIPTION OF AN OBJECT BY MEANS OF A PREDICATE CALCULUS LANGUAGE

Tatiana Kosovskaya

***Abstract**: A problem of multi-agent description of a complex object is under consideration. It is supposed that an object may be described in the terms of its parts properties and relations between these parts. But every agent has only a part of the object description and does not know the true names of elements and gives them names arbitrary. An algorithm solving the posed problem is described and the upper bound of its steps is proved.*

***Keywords**: multi-agent description of an object, predicate calculus, partial deduction, computational complexity of an algorithm.*

***ACM Classification Keywords**: I.2.4 Artificial Intelligence - Knowledge Representation Formalisms and Methods - Predicate logic; I.5.1 Pattern Recognition Models – Deterministic; F.2.2 Nonnumerical Algorithms and Problems – Complexity of proof procedures.*

Introduction

The use of a predicate calculus language for the Artificial Intelligence problems was proposed in the middle of the XX century (see, for example, [Duda, Nilsson]) and up to now is offered in theoretical papers [Russel]. Just a predicate calculus language allows adequately describe a complex object characterizing by properties of its parts and relations between them.

Nevertheless the difficulty of a practical implementation of such an approach is connected with NP-hardness of many problems appearing in the frameworks of this approach. The proven upper bound of an algorithm solving such a problem allows for every problem to find such restrictions upon input data which essentially decrease the number of the algorithm run steps. That's why upper the upper bound of the offered algorithm number of steps are proved bellow.

A problem of multi-agent description of a complex object is under consideration in the presented paper. It is supposed that every agent has only a part of an investigated object description. Moreover, she does not know the true names of elements and gives them names arbitrary. It is similar to the parable about tree blind men who feel an elephant. To overcome such a paradox, it is supposed that every two agents have information concerning some common part of an object. The main difficulty in this problem is to find and identify these parts.

Multi-agent description problem setting

Let an investigated object is represented as a set of its elements $\omega = \{\omega_1, ..., \omega_t\}$ and is characterized by the set of predicates $p_1, ..., p_n$, every of which is defined on the elements of ω and gives properties of these elements and relations between them.

Information (description) of the object is an elementary conjunction of atomic formulas with predicates $p_1, ..., p_n$.

There are *m* agents $a_1, ..., a_m$ which can measure some values for some predicates of some elements of ω. The agent a_j does not know the true number of the ω elements and suppose that she deals with the object $\omega^j = \{\omega_1^j, ..., \omega_{tj}^j\}$. That is the agent a_j has the information $I_j(\omega_1^j, ..., \omega_{tj}^j)$ in the form of elementary conjunction of atomic formulas.

It is required to construct the description of ω $I(\omega_1, ..., \omega_t)$.

As every agent uses her own notifications for the names of the object elements, it is needed to find all common up to the names of arguments sub-formulas of the informations $I_j(\omega_1^j, ..., \omega_{tj}^j)$ ($j = 1, ..., m$) and their unifiers, i.e. such substitutions for the argument names that the extracted pairs of sub-formulas are identical.

For example, two formulas $A = p(a,b)\ \&\ p(b,a)\ \&\ q(b,a,c)$ and $B = p(b,a)\ \&\ p(b,d)\ \&\ q(a,b,d)$ have common up to the names of arguments sub-formula. If one substitute arguments *u, v, w* instead of the constants
a, b, c in the formula A and substitute arguments v, *u, w* instead of the constants *a, b, d* in the formula B we receive formulas $C1 = p(u,v)\ \&\ p(v,u)\ \&\ q(v,u,w)$ and $C2 = p(u,v)\ \&\ p(u,w)\ \&\ q(v,u,w)$ respectively. The formula $C = p(u,v)\ \&\ q(v,u,w)$ is a common up to the names of arguments sub-formula of *A* and *B* with unifiers λ_{CA} (substitution of *u, v, w* instead of *a, b, c*) and λ_{CB} (substitution of v, *u, w* instead of *a, b, d*).

A notion of partial deduction (introduced in [Kosovskaya, 2009]) allows to extract a maximal common up to the names of arguments sub-formula of *A* and *B* and to find their unifiers. During the process of partial deduction instead of the proof of $A(\boldsymbol{\omega}) \Rightarrow B(\boldsymbol{\tau})$ we search such a maximal (up to the names of arguments) sub-formula $B'(\mathbf{x}')$ of the formula $B(\mathbf{x})$ that $A(\omega) \Rightarrow \exists \mathbf{x}'_{\neq}\ B'(\mathbf{x}')$. In the process of this sequent proof the unifier of $B'(\mathbf{x}')$ and a sub-formula of $A(\boldsymbol{\omega})$ will be found. The notation $A(\boldsymbol{\omega}) \Rightarrow_P$

□$B(\tau)$ will be used for the partial deduction process. A detailed description of the partial deduction checking and the search of the unifier is in [Kosovskaya, 2014].

Algorithm of multi-agent description

Let every agent a_j has information I_j about the described object ω ($j = 1, \ldots, m$). To constract a description of ω the following algorithm is offered.

1. Change all constants in $I_1, \ldots, I_m$ by variables in such a way that different constants are changed by different variables and the names of variables in I_i and I_j ($i \neq j$) does not coinside.
2. For every pair of elementary conjunctions I_i and I_j ($i = 1, \ldots, m-1, j = i+1, \ldots, m$) check partial deduction $I_i \Rightarrow_P I_j$ with the extraction of their common up to the names of arguments sub-formula C_{ij} and unifiers $\lambda_{i,ij}$ and $\lambda_{j,ij}$. Every argument of C_{ij} has a unique name.
3. For every pair i and j ($i > j$) check if I_i and I_j contain a contradictory pair of atomic formulas or two sub-formulas which can not be satisfiable simultaneously (for example, "*x is green*" and "*x is red*"). If such a contradiction is established then delete from C_{ij} atomic formulas containing the variables which are in the contradictory sub-formulas. Change the unfiers.
4. For every i identify the variables in C_{ij} ($i \neq j$) which are substituted in I_i and I_j instead of the same variable. The names of the identified variables are changed in unifiers by the same name.
5. With the use of the unifiers obtained in 2 – 4 change the names of variables in $I_1, \ldots, I_m$.
6. Write down the conjunction $I_1 \& \ldots \& I_m$ and delete the repeating atomic formulas.

Upper bound of the algorithm run steps

To estimate the number of the algorithm run steps we estimate every item of the algorithm.

1. The change all constants in $I_1, \ldots, I_m$ requires not more than $\sum_{j=1}^m ||I_j||$ «steps».
2. The checking of partial deduction $I_i \Rightarrow_P I_j$ requires $O(t_i^{t_j} \cdot 2^{||I_i||})$ «steps» for an exhaustive algorithm and $O(||I_i||^{||I_j||} \cdot ||I_i||^3)$ «steps» for an algorithm based on the derivation in the predicate calculus. (These estimates are proved in [Kosovskaya, 2011].) It is needed to summarize the above estimates for $i = 1, \ldots, m-1, j = i, \ldots, m$. So we have $O(t^t\ 2^{||I||} m^2)$ «steps» for an exhaustive algorithm and $O(||I||^{||I||+3}\ m^2)$ «steps» for an algorithm based on the derivation in the predicate calculus. Here t and $||I||$ are respectively the maximal numbers of variables and atomic formulas in I_j ($j = 1, \ldots, m$).
3. Consistency checking of the formulas I_i and I_j requires $||I_i||\ ||I_j||$ «steps». This item of the algorithm requires not more than $\sum_{i=1}^m (m-i) ||I_i||$ «steps» that is $O(m^2 ||I||)$ «steps».
4. For every i identification of the variables in C_{ij} ($i > j$) consists in the comparison of the replaced part of the unifiers $\lambda_{i,ij}$ and $\lambda_{j,ij}$. It requieres not more than $(m-i)t_i^2$ «steps». Summarizing it for $i = 1, \ldots, m$ we have not more than $\sum_{i=1}^m (m-i)t_i^2 = O(m^2 t^2)$ «steps».

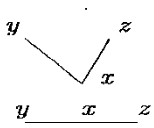
5. The number of «steps» required for the changing of the names of variables in $I_1, \ldots, I_m$ is linear under $\sum_{i=1}^{m} ||I_i|| = O(m\,||I||)$ «steps».
6. The number of «steps» required for the deleting of the repeated conjunctive terms is not more than $\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} ||I_i||\,||I_j|| = O(m^2 t^2)$ «steps».

The number of the algorithm run steps is $O(t^t m^2 ||I||^2)$ for an exhaustive algorithm and $O(||I||^{||I||+3} m^2)$ for an algorithm based on the derivation in the predicate calculus.

The analysis of the received estimation shows that the main contribution to it is made by the summarized number of partial deduction checking (item 2).

Examples

Consider an example of description of a contour image of a "box" by 3 agents in the terms of three predicates V and L represented on the Figure 1.[1]



$V(x,y,z) \Leftrightarrow$ "$\angle yxz < \pi$"

$L(x,y,z) \Leftrightarrow$ "*x belongs the segment (y,z)*"

Figure 1. Initial predicates

These predicates characterize the position of the node x relatively the nodes y, z and have the following properties: $V(x,y,z) \Leftrightarrow \neg V(x,z,y) \vee L(x,y,z)$ (and hence $V(x,y,z)$ & $\neg V(x,z,y)$ is a contradiction) and $L(x,y,z) \Leftrightarrow L(x,z,y)$.

Every agent has a description of one of the fragment represented on the Figure 2.

[1] The extracting of the maximal common up to the names of arguments sub-formula of two elementary conjunctions, the search of their unifier and expression the formulas through the extracted sub-formulas was made with the use of a software support implemented by a student of faculty of mathematics and mechanics of Sankt-Petersburg State University Petrov D.A.

As the arguments in the descriptions of these fragments are very important, they are written down explicitly.

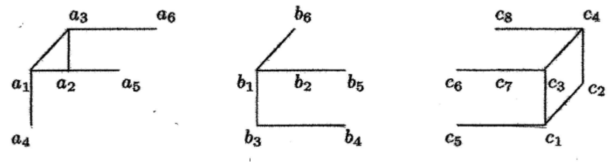


Figure 2. Fragments of the image received by three agents

According to the item 1 of the algorithm all constants in the fragment descriptions are replaced by variables in such a way that different constants are changed by different variables and the names of variables in I_i and I_j ($i \neq j$) does not coincide. After this the fragment descriptions have the form:

I1(x1,...,x6) = V(x1,x2,x4) & V(x1,x5,x4) & V(x1,x3,x2) & V(x1,x3,x5) & V(x1,x3,x4) & V(x2,x1,x3) & V(x2,x3,x5) & V(x3,x2,x1) &V(x3,x6,x2) & V(x3,x6,x1) & L(x2,x1,x5),

I2(y1,...,y6) = V(y3,y1,y4) & V(y1,y2,y3) & V(y1,y5,y3) & V(y1,y6,y2) & V(y1,y6,y5) & V(y1,y6,y3) & L(y2,y1,y5),

I3(z1,...,z8) = V(z1,z5,z3) & V(z1,z3,z2) & V(z1,z5,z2) & V(z3,z1,z7) & V(z3,z1,z6) & V(z3,z7,z4) & V(z3,z6,z4) & V(z3,z4,z1) & V(z4,z2,z3) & V(z4,z3,z8) & V(z4,z2,z8) & L(z7,z6,z3).

According to the item 2 of the algorithm check pairwise partial deduction $I_i \Rightarrow_P I_j$.

Maximal common up to the names of arguments sub-formula of *I1(x1,...,x6)* and *I2(y1,...,y6)* is *C12(u0,...,u4)* in the form

C12(u0,...,u4) = V(u0,u1,u2) & V(u0,u3,u2) & V(u0,u4,u1) & V(u0,u4,u3) & V(u0,u4,u2) & L(u1,u0,u3).

It has unifiers $\lambda_{I1,C12}$ – substitution of *u0, u1, u4, u2, u3* instead of *x1, x2, x3, x4, x5* respectively and $\lambda_{I2,C12}$ – substitution of *u0, u1, u2, u3, u4* instead of y*1, y2, y3, y5, y6* respectively. Besides,

I1(u0,u1,u2,u3,u4,x6) =

= V(u1,u0,u4) & V(u1,u4,u3) & V(u4,u1,u0) & V(u4,x6,u1) & V(u4,x6,u0) & C12(u0, ..., u4),

I2(u0,u1,u2,y4,u3,u4) = V(u2,u0,y4) & C12(u0,...,u4).

Maximal common up to the names of arguments sub-formula of *I2(y1,...,y6)* and *I3(z1,...,z8)* is *C23(v0,v2,v4,v5,v6,v7)* in the form

C23(v0,v2,v4,v5,v6,v7) = V(v6,v2,v7) & V(v2,v4,v6) & V(v2,v5,v6) &V(v2,v0,v4) & V(v2,v0,v5).

It has unifiers $\lambda_{I2,C23}$ *– substitution of v2, v4, v6, v7, v5, v0 instead of y1, y2, y3, y4, y5, y6* respectively *and* $\lambda_{I3,C23}$ – *substitution of v0, v2, v6, v5, v4, v7 instead of z1, z3, z5, z6, z7, z8* respectively. Besides,

I2(v2,v4,v6,v7,v5,v0) = V(v2,v0,v6) & L(v4,v2,v5) & C23(v0,v2,v4,v5,v6,v7),
I3(v0,z2,v2,v6,z5,v5,v4,v7) =

= V(v2,v6,v0) & V(v0,z5,v2) & V(v0,v2,z2) & V(v0,v5,z2) & V(v6,z2,v2) & V(v6,v2,v7) & L(v4,v5,v2) &

& C23(v0,v2,v4,v5,v6,v7).

As *I2(v2,v4,v6,v7,v5,v0)* contains *V(v2,v0,v6)* and I3(v0,z2,v2,v6,z5,v5,v4,v7) contains V(v2,v6,v0) and according to the definition of the predicate *V* the formula *V(x,y,z) & V(x,z,y)* is a contradiction so substitutions with this unifiers can not give a consistent description of the object. After deleting from *I2(y1,...,y6)* and *I3(z1,...,z8)* the variables *y1* and *z3* respectively a new maximal common up to the names of arguments their sub-formula *C'23(v0,v2,v4,v5,v6,v7)* in the form

C'23(v0,v1,v2) = L(v1,v0,v2)

will be received with the unifiers $\lambda_{I2,C'23}$ – substitution *of v0, v1, v2* instead of *y1, y2, y3* respectively and $\lambda_{I3,C'23}$ – substitution of *v2, v0, v1* instead of *z3, z6, z7* respectively. Besides,

I2(v0,v1,v2,y4,y5,y6) =

= V(v2,v0,y4) & V(v0,v1,v2) & V(v0,y5,v2) & V(v0,y6,v1) &V(v0,y6,y5) & V(v0,y6,v2) & C'23(v0,v1,v2),
I3(z1,z2,v2,z4,z5,v0,v1,z8) =

= V(z1,z5,v2) & V(z1,v2,z2) & V(z1,z5,z2) & V(v2,z1,v1) & V v2,z1,v0) & V(v2,v1,z4) & V(v2,v0,z4) & V(v2,z4,z1) & V(z4,z2,v2) & V(z4,v2,z8) & V(z4,z2,z8) & C'23(v0,v1,v2).

Maximal common up to the names of arguments sub-formula of *I1(x1,...,x6)* and *I3(z1,...,z8)* is *C13(w0, ...,w6)* in the form

C13(w0, ...,w6) = V(w2,w4,w6) & V(w2,w5,w6) & V(w2,w0,w4) & V(w2,w0,w5) & V(w0,w1,w2).

It has unifiers $\lambda_{I1,C13}$ *– substitution of w2, w4, w0, w6, w5, w6 instead of x1, x2, x3, x4, x5, x6* respectively *and* $\lambda_{I3,C13}$ – *substitution of w0, w2, w6, w1, w5, w2 instead of z1, z3, z4, z5, z6, z7* respectively. Besides,

I1(w2,w4,w0,w6,w5,w1) = V(w2,w0,w6) & V(w0,w1,w4) & V(w0,w4,w2) & L(w2,w4,w5) & C13(w0,...,w6),
I3(w0,z2,w2,w6,w1,w5,w4,z8) =

= V(w0,w2,w3) & V(w0,w1,w3) & V(w2,w6,w0) & V(w6,w3,w2) & V(w6,w2,w7) & V(w6,w3,w7) &

& C13(w0,...,w6).

As *I1(w2,w4,w0,w6,w5,w1)* contains *V(w2,w0,w6), I3(w0,z2,w2,w6,w1,w5,w4,z8)* contains *V(w2,w6,w0)* and according to the definition of the predicate *V* the formula *V(x,y,z) & V(x,z,y)* is a contradiction so substitutions with this unifiers can not give a consistent description of the object. After deleting from

I1(x1,...,x6) and I3(z1,...,z8) the variables *x1* and *z3* respectively a new maximal common up to the names of arguments their sub-formula *C'13(w0,w1,w2)* in the form

C'13(w0,w1,w2) = L(w1,w0,w2)

will be received with the unifiers $\lambda_{I1,C'13}$ – substitution *of* w*0, w1, w2* instead of *x1, x2, x5* respectively and $\lambda_{I3,C'13}$ – substitution of *w2, w1, w0* instead of *z3, z4, z5* respectively.

Besides,

I1(w0,w1,x3,x4,w2,x6) =

= V(w0,w1,x4) & V(w0,w2,x4) & V(w0,x3,w1) & V(w0,x3,w2) & V(w0,x3,x4) & V(w1,w0,x3) &

& V(w1,x3,w2) & V(x3,w1,w0) & V(x3,x6,w1) & V(x3,x6,w0) & C'13(w0,w1,w2),

I3(z1,z1,w2,w1,w0,z6,z7,z8) =

= V(z1,w0,w2) & V(z1,w2,z2) & V(z1,w0,z2) & V(w2,z1,z7) & V(w2,z1,z6) & V(w2,z7,w1) &

& V(w2,z6,w1) & V(w2,w1,z1) & V(w1,z2,w2) & V(w1,w2,z8) & V(w1,z2,z8) & C'13(w0,w1,w2).

According to the item 4 of the algorithm we identify new variables substituted instead of the same initial variable. That is we identify the following variables

u0 and *w0* (are substituted instead of the variable *x1*),
u1 and *w1* (are substituted instead of the variable *x2*),
u2 and *w2* (are substituted instead of the variable *x4*),
u0 and *v0* (are substituted instead of the variable *y1*),
u1 and *v1* (are substituted instead of the variable *y2*),
u2 and *v2* (are substituted instead of the variable *y3*),
v0 and *w0* (are substituted instead of the variable *z6*),
v1 and *w1* (are substituted instead of the variable *z3*),
v2 and *w2* (are substituted instead of the variable *z7*).

The identified variables denote as $\alpha0, \alpha1, \alpha2.$ So we have the equalities *u0 = v0 = w0 =* $\alpha0,$ *u1 = v1 = w1 =* $\alpha1,$ *u2 = v2 = w2 =* $\alpha2.$

As a result we have the following descriptions of the fragments

$I1(I1(\alpha0,\alpha1,u4,u2,\alpha2,x6) =$

$= V(\alpha0,\alpha1,u2) \& V(\alpha0,\alpha2,u2) \& V(\alpha0,u4,\alpha1) \& V(\alpha0,u4,\alpha2) \& V(\alpha0,u4,u2) \& V(\alpha1,\alpha0,u4) \&$

$\& V(\alpha1,u4,\alpha2) \& L(\alpha1,\alpha0,\alpha2) \& V(x3,\alpha1,\alpha0) \& V(u4,x6,\alpha1) \& V(u4,x6,\alpha0),$

$I2(\alpha0,\alpha1,u2,y4,\alpha2,u4) =$
$= V(u2,\alpha0,y4) \& V(\alpha0,\alpha1,u2) \& V(\alpha0,\alpha2,u2) \& V(\alpha0,u4,\alpha1) \& V(\alpha0,u4,\alpha2) \& V(\alpha0,u4,u2) \&$
$\& L(\alpha1,\alpha0,\alpha2),$

$I3(z1,z2,\alpha2,z4,z5,\alpha0,\alpha1,z8) =$

$= V(z1,z5,\alpha2) \& V(z1,\alpha2,z2) \& V(z1,z5,z2) \& V(\alpha2,z1,\alpha1) \& V(\alpha2,z1,\alpha0) \& V(\alpha2,\alpha1,z4) \&$

$\& V(\alpha2,\alpha0,z4) \& V(\alpha2,z4,z1) \& V(z4,z2,\alpha2) \& V(z4,\alpha2,z8) \& V(z4,z2,z8) \& L(\alpha1,\alpha0,\alpha2).$

Their conjunction

$V(\alpha0,\alpha1,u2) \& V(\alpha0,\alpha2,u2) \& V(\alpha0,u4,\alpha1) \& V(\alpha0,u4,\alpha2) \& V(\alpha0,u4,u2) \& V(\alpha1,\alpha0,u4) \& V(\alpha1,u4,\alpha2) \& V(x3,\alpha1,\alpha0) \& V(u4,x6,\alpha1) \& V(u4,x6,\alpha0) \& V(u2,\alpha0,y4) \& V(z1,z5,\alpha2) \& V(z1,\alpha2,z2) \& V(z1,z5,z2) \& V(\alpha2,z1,\alpha1) \& V(\alpha2,z1,\alpha0) \& V(\alpha2,\alpha1,z4) \& V(\alpha2,\alpha0,z4) \& V(\alpha2,z4,z1) \& V(z4,z2,\alpha2) \& V(z4,\alpha2,z8) \& V(z4,z2,z8) \& L(\alpha1,\alpha0,\alpha2)$

allows to "stick together" the images of fragments according to the same variable. The image corresponding to the result of "sticking" is presented on the Figure 3.

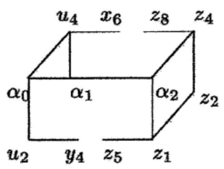


Figure 3. Image corresponding to the result of "sticking"

If a description of the investigated object is presented in the database it may be found according the principle "the nearest neighbor" with the use of metric for predicate formulas presented in [Kosovskaya, 2012].

Conclusion

An algorithm solving a rather complicated problem of multi-agent description of a complex object in the terms of the predicate calculus language with the condition that different agents may give different names to the same elements of the object is presented in the paper. The length of such an object description in the terms of the propositional calculus language is exponential in comparison with that in the terms of the predicate calculus language [Rassel]. It explains the exponential upper bound of the algorithm number of steps. Moreover, it may be easily proved that the problem under consideration is NP-hard.

The analysis of the received estimations allows to formulate restrictions upon the initial predicates for decreasing of the practical time of the algorithm run. For example, if we deal with a great number of initial predicates every of which has very small number of occurrences in the object description then the practical time of the algorithm run decreases.

Acknowledgements

The paper is published with financial support of RFBR grant 14-08-01276.

Bibliography

[Duda] Duda R.O. and Hart P.E., "Pattern Classification and Scene Analysis", A Wiley-Interscience Publication, John Wiley \& Sons, New York London Sydney Toronto, 1973.

[Kosovskaya, 2009] T. M. Kosovskaya. "Partial deduction of a predicate formula as an instrument for recognition of an object with incomplete description", In: Vestnik of St. Petersburg University. Ser. 10, 2009, No. 1, pp. 77 – 87 (In Russian)

[Kosovskaya, 2011] T. Kosovskaya, "Discrete Artificial Intelligence Problems and Number of Steps of their Solution" // In: International Journal "Information Theories and Applications", Vol. 18, Number 1, 2011, pp. 93 – 99

[Kosovskaya, 2012] Kosovskaya T., "Distance between objects described by predicate formulas" // In: International Book Series. Informational Science and Computing. Book 25. Mathematics of Distances and Applications (Michel Deza, Michel Petitjean, Krasimir Markov (eds)), ITHEA-Publisher, Sofia, Bulgaria, 2012, pp. 153 – 159

[Kosovskaya, 2014] Kosovskaya T., "Construction of Class Level Description for Efficient Recognition of a Complex Object" // International Journal "Information Content and Processing", Vol. 1, No 1, 2014, pp. 92 – 99

[Nilsson] Nilsson N.J., "Problem-solving methods in artificial intelligence", mcgraw-hill book company, New York, 1971.

[Russel] Russel S.J. and Norvig P., "Artificial Intelligence. A Modern Approach.", Pearson Education, Inc. 2003.

Authors' Information

Tatiana Kosovskaya *– Dr., Professor of St. Petersburg State University, University av., 28, Stary Petergof, St .Petersburg, 198504, Russia; e-mail: kosovtm@gmail.com*

Major Fields of Scientific Research: Logical approach to artificial intelligence problems, theory of complexity of algorithms

ON THE MECHANIZATION OF KLEENE ALGEBRA IN FORMAL LANGUAGE ANALYZER

Dmitry Cheremisinov

Abstract: *Pattern matching is the technique of searching a text string based on a specific search pattern. The pattern specified by the regular expression forms the basis for building a variety of formal language texts converters. Kleene algebra or the algebra of regular events is an algebraic system that captures properties of several important structures arising in computer science like automata and formal languages, among others. Regular expressions are formulas of Kleene algebra. In this paper we present a formalization of regular expressions as Kleene algebra in the formal language analyzer. It is proposed to change the traditional model of the language parser by pattern matching based on the finite state machine into the algebra of patterns with side effects. The proposed deterministic semantic of regular expression eliminates the need to switch from the regular expression engine and user code execution environment and back again.*

Keywords: *Formal language parser, regular expression, Kleene algebra, finite automaton, deterministic semantic of regular expression.*

ACM Classification Keywords: *I. Computing Methodologies: I.1 SYMBOLIC AND ALGEBRAIC MANIPULATION: I.1.3 Languages and Systems, Special-purpose algebraic systems.*

Introduction

Regular expressions [Friedl, 2006] are often used in practice in order to build programs, which are text analyzers. It is quite common to generate useful and efficient parsers for programming languages from a formal grammar. It is also quite common for programmers to avoid such tools when making parsers for simple computer languages, such as file formats and communication protocols. Such languages are often regular and tools for processing the context-free languages are viewed as too heavyweight for the purpose of parsing regular languages. The extra run-time effort required for supporting the recursive nature of context-free languages is wasted.

Processor of regular expressions is a parser based on a deterministic finite automaton. This machine may be represented as a static data of the program in the form of the output/transitions table. The table-controlled analyzers are used in Perl, Python, Emacs, Tel and Net. The processor of regular expressions does not "catch" the subexpression of the original regular expression, because an indication of recognition of a regular language sentence is transition to the final state.

Unlike the problem of recognizing language that traditionally is considered in the theory of formal languages, the task of language analyzer is to build the structure of the analyzed text, when it is parsed. A program that attempts to verify written text for grammatical correctness is a grammar checker. The recognition algorithm has the form of grammar. To construct a grammar analyzer it is necessary to add steps that form a data structure that represents the result of the analysis. Regular expressions describe regular languages. They have the same expressive power as regular grammars. Actions to recognize parts of the text alternate with actions of constructing the parse results in parser algorithm.

The structure of the analyzed text is reflected in the structure of a regular expressions. The way to use the information about this structure is to include the action code in the analysis process. The action code constructs the results of analysis. The need for interleaving of the analysis and action puts restrictions on how to include action code in the parsing algorithm. Since the actions must be performed after the transition of the machine to the final state, the actions that should be performed after the recognition of subexpressions may be included in the program only by splitting a regular expression into smaller units. The more actions are included in the analysis process, the fewer benefits from a regular expression processor as a software tool, since it reduces the code generated by a regular expression and it augments the percentage of software "glue".

It is proposed to substitute the traditional model of the operation of pattern matching based on the finite state machine model for the model of the pattern algebra. Regular expression language becomes deterministic in the interpretation of the pattern algebra, ensuring the inclusion of the action code not only at the end of the expression, but also after subexpressions.

The regular expression language

Regular expressions are represented as a set of possible formulas of a Kleene algebra. So, Kleene algebra is formal semantics, or interpretation of regular expressions as a formal language. We now recall some basic definitions of formal languages and Kleene algebra that we need throughout the paper. For further details one can use the works of Hopcroft et al. [HMU, 2000] and Kozen [Kozen, 1997].

An alphabet Σ is a nonempty set of symbols. A word w over an alphabet Σ is a finite sequence of symbols from Σ. The empty word is denoted by ε and the length of a word w is denoted by $|w|$. The concatenation " $\cdot$ " of two words w_1 and w_2 is a word $w = w_1 \cdot w_2$ obtained by juxtaposing the symbols of w_2 after the last symbol of w_1. The set Σ^* contains all words over the alphabet Σ. The triple $(\Sigma^*, \cdot, \varepsilon)$ is a monoid.

A language L is subset of Σ^*. If L_1 and L_2 are two languages, then $\mathrm{L}_1 \cdot \mathrm{L}_2 = \{\mathrm{xy} | \mathrm{x} \in \mathrm{L}_1$ and $\mathrm{y} \in \mathrm{L}_2\}$. The operator $\cdot$ is often omitted. For $\mathrm{n} \geq 0$, the n^{th} power of a language L is inductively defined by $\mathrm{L}^0 = \{\}$, $\mathrm{L}^{\mathrm{n}} = \mathrm{LL}^{\mathrm{n}-1}$. The Kleene's star L^* of a language L, is $\bigcup_{n \geq 0} \mathrm{L}^n$. A regular expression (r.e.) r over Σ represents a regular language $\mathrm{L}(r) \subseteq \Sigma^*$ and is inductively defined like that: $\emptyset$ is a r.e. and $\mathrm{L}(\emptyset) = \emptyset$; ε is a r.e. and $\mathrm{L}(\varepsilon) = \{\}$; $a \in \Sigma$ is a r.e. and $\mathrm{L}(a) = \{a\}$; if r_1 and r_2 are r.e., $(\mathrm{r}_1 + \mathrm{r}_2)$, (rr) and $(\mathrm{r}_1)^*$ are r.e., respectively with $\mathrm{L}((\mathrm{r}_1 + \mathrm{r}_2)) = \mathrm{L}(\mathrm{r}_1) \cup \mathrm{L}(\mathrm{r}_2)$, $\mathrm{L}((\mathrm{r}_1\mathrm{r}_2)) = \mathrm{L}(\mathrm{r})\mathrm{L}(\mathrm{r})$ and $\mathrm{L}((\mathrm{r}_1)^*) = \mathrm{L}(\mathrm{r}_1)^*$. We adopt the usual convention that "$*$" has precedence over $\cdot$, and "$\cdot$" has higher priority than "$+$", and we omit outer parentheses. Let RegExp be the set of regular expressions over Σ, and let Reg_Σ be the set of regular languages over Σ. Two regular expressions r_1 and r_2 are equivalent if $(\mathrm{r}_1) = \mathrm{L}(\mathrm{r}_2)$, and we write (the equation) $\mathrm{r}_1 = \mathrm{r}_2$. The equational properties of regular expressions are axiomatically captured by a KA, normally called the algebra of regular events, after the seminal work of S.C. Kleene [Kleene,1956].

A $\mathcal{K} = (\mathrm{K}, 0, 1, +, \cdot, *)$ is an algebraic structure such that $(\mathrm{K}, 0, 1, +, \cdot)$ is an idempotent semiring and where the operator "$*$" (Kleene's star) is characterized by a set of axioms. We also assume a relation $\leq$ on K, defined by $\mathrm{a} \leq \mathrm{b} \Leftrightarrow_{\mathrm{def}} \mathrm{a} + \mathrm{b} = \mathrm{b}$, for any $\mathrm{a}, \mathrm{b} \in \mathrm{K}$.

Let S be a chain (a binary relation on Σ^*, which is transitive, antisymmetric, and total) of words w over an alphabet Σ. For chain S we can define the open interval $(\mathrm{a}, \mathrm{b}) = \mathrm{X}$. The constituency set C is the set of intervals in S. This set of intervals contains S and each word $\mathrm{w} \in \mathrm{S}$ as its elements, and is constructed this way, so any two intervals belonging to C either do not intersect or one of them is contained in the other. The elements of such sets are called constituents. The constituent x dominates the constituent y, if y is a part of x and y differs from x. Constituents of a constituency set of regular expression r are symbols from Σ and operation symbols from the set $\langle \cdot, +, \cdot \star, 1, 0 \rangle$. Given a constituency set C, the analysis process divides up a word into major parts or immediate constituents, and these constituents are in turn divided into further immediate constituents. The process continues until irreducible constituents are reached. The end result of analysis is presented in a tree form that reveals the hierarchical immediate constituent structure of the word at hand. This is a parse tree of regular expression. Regular expressions can express the regular languages, exactly the class of languages accepted by deterministic finite automata.

Let R be a regular expression. We can construct a finite automaton M with $\mathrm{L}(\mathrm{M}) = \mathrm{L}(\mathrm{R})$ recursively. The idea is to use the fact that the set of languages of finite automata is closed under union, concatenation, and Kleene star operations. In general case, R is the constant, either $\mathrm{R} = \mathrm{x}$ for $\mathrm{x} \in \Sigma$, $\mathrm{R} = \varepsilon$ or $\mathrm{R} = \emptyset$. In all cases, $\mathrm{L}(\mathrm{R})$ is finite. Hence, there exists a trivial finite automaton for $\mathrm{L}(\mathrm{R})$.

Otherwise, R is an operation applied to one or two smaller expressions. Either $\mathrm{R} = \mathrm{R}_1 \vee \mathrm{R}_2$, $\mathrm{R} = \mathrm{R}_1\mathrm{R}_2$, or $\mathrm{R} = \mathrm{R}_1 *$. Since R_1 and R_2 are smaller regular expressions, we can construct automata M_1

and M_2 with $L(M_1) = L(R_1)$ and $L(M_2) = L(R_2)$. Then there exist finite automata for the languages $L(M_1) \cup L(M_2)$, $L(M_1)L(M_2)$ and $L(M_1)*$. Therefore, there exists a finite automaton for $L(R)$. We can construct a finite (nondeterministic) automaton for $L(R)$ by conversion constituency set of R.

The finite automaton is $M = (\Sigma, Q, q_0, T, P)$, where Σ is an alphabet; Q – a finite set of states; $q_0 \in Q$ is the initial state; $T \subset Q$ – a set of terminal conditions; P – the transition function defined by the set of rules in the form of $q_i a_k q_j$, where q_i and q_j are the states, the input symbol $a_k \in \Sigma$ or it is the empty symbol ε. A finite automaton is deterministic finite automaton (DFA), if each of its transitions is uniquely determined by its source state and input symbol, and reading an input symbol is required for each state transition. A nondeterministic finite automaton (NFA) does not need to obey these restrictions. The transitions without consuming an input symbol are called ε-transitions. ε-transitions appear in the component automata, when constituents $R_1 \vee R_2$ or $R*$ are transformed.

An analysis state of the word $v \in \Sigma^*$ is an ordered pair (v, i), with i being an integer $0 \leq i \leq |v|$; $|v|$ is the length of the word. An analysis state (v, i) is the representation of the word in hand in the form of the concatenation $v = br$, $|b| = i$. A transition rule $q_i a_k q_j$ of an automaton M can be applicable into analysis state (v, i), if it can be represented as a concatenation $v = ba_k r, |b| = i$ and the current state of the machine is q_i. If the automaton M is the NFA then its current state is a set of states and q_i must enter into the set. The application of the transition rule transfers the automaton M to the state q_j (if automaton M is the NFA then q_j becomes an element of the current state) and the current analysis state becomes $(v, i + 1)$. The automaton M is applicable to the analysis state (v, i) if there is a rule for the initial state q_0 applicable in the analysis state (v, i), and after applying all possible transition rules the automaton M is in a terminal state. Initial analysis state (v, l) and the analysis state (v, m), when the machine is in a terminal state, give a representation of the word v as the concatenation $v = bxt, |b| = l, |bx| = m$. The word x is recognized as a part of v by a regular expression R if $L(M) = L(R)$ for the automaton M. The set of all words that are recognized by a regular expression R generates a regular language.

In this interpretation (semantic) of the regular expression language a regular expression is partial function from the analysis states of word in hand. Text analysis with the interpretation of the analysis operation as a partial function from the set of analysis states is the analysis by patterns [Aho, 1990]. String searching algorithms try to find a place where a string called pattern is found within a larger string or text. In most programming languages there is the realization of such an operation, in which the pattern is recognizable word itself (regular expression constructed by concatenation only). The pattern search operation is a lexicographic enumeration of analysis states for the purpose of finding (one or several) occurrences of a pattern. Boost.Regex [Boost, 2002] allows using regular expressions in C++

programs into pattern search operations. As the Boost library is a part of the standard library since C++11 programmers don’t depend on Boost.Regex if their development environment supports C++11.

Nondeterministic nature of standard interpretation of the regular expression language

The process of applying transition rules to the automaton M may be fixed in the form of a parse tree in the same way as it is done by parsing. The set of the transition rules of the automaton M is a grammar in which the alphabet of non-terminal symbols is a set of states. A parse tree is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar. But the set of the transition rules of the nondeterministic automaton M is not a context-free grammar. An internal vertex of the automaton parsing process tree is labeled by the current state (a current state of a non-deterministic machine is a set of nonterminal symbols). If such an internal vertex is split then the terminal vertexes appear, which are dead-ends of analysis. In this case the non-determinism of the standard semantic of regular expressions is manifested. The parse tree cannot be built by linking actions with machine instructions (parsing algorithm does not "catch" subexpressions).

There exists an algorithm (the power set construction) that can transform the NFA M into a more complex DFA with identical functionality. The set of the transition rules of a deterministic automaton is a context-free grammar and a parse tree can be built by linking actions with transition rules of DFA. However, this set of transition rules has no structural similarity with the constituency set of regular expression in hand, and therefore the structure of the regular expression cannot be used during the analysis of a text.

We can use language formed by a limited subset of regular expressions for the analysis of sub-expressions. Regular expressions of this limited subset are transformed into deterministic automata. Some transition rules with an empty input symbol can be deleted preserving equivalence of automata. However, this approach significantly reduces the figurative possibility of regular expression language and significantly increases the risk of programming errors, as the procedure for establishing the properties of determinism is not trivial. This approach is proposed for use in the Ragel State Machine Compiler [Thurston, 2007].

The pattern algebra

A partial function is a function that is defined only on a part of its domain. A McCarthy conditional expression [McCarthy, 1960] has the form $(p_1 \to f_1; p_2 \to f_2; \ldots; p_n \to f_n)$ defining a partial function h, coinciding with one of the functions f_i, where the number i satisfies the following condition:

$$\exists \mathrm{i}(p_i(x) \vee \forall j\left(\left(j < i \Rightarrow \neg p_j\left(((x))\right)\right)\right);$$

where the symbol ¬ denotes the inverse of logical value. If such i does not exist the function h is not specified. Conditional expressions are a device for expressing the dependence of quantities on propositional quantities. Here variables p_i's correspond to propositional expressions and the variables f_i's are expressions of any kind. A propositional expression (predicate) is an expression whose permissible values are T (for truth) and F (for falsity). The rule for determining whether the value of a McCarthy conditional expression is defined can be determined in the following way. Examine the p's from left to right. If a p whose value is T is encountered before any p with undefined value is encountered, then the value of the conditional expression equals to the value of the corresponding f (if it is defined). If any undefined p is encountered before a true p, or if all p's are false, or if the f corresponding to the first true p is undefined, then the value of the conditional expression is undefined.

Let f_i и p_i are symbols of some functions and predicates. The partial conversion function f, specified on the set of analysis states to parse a string, is the pattern, if for any states $\alpha=(w,j),\beta=(v,i),\beta=f(\alpha)$ and $w=v, j\le i$. Let $p(f,\alpha)$ is the predicate whose value is T if the pattern f is defined on the analysis state α. If we have for patterns $f_1=f_2$ then $p(f_1)=p(f_2)$.

The basic relationship describing a function is that of application. Let a function that applies functions to arguments is called as an apply function. This establishes a one-to-one correspondence between functions of two variables and functions returning functions, which we know under the name of currying. We will use the infix notation for apply function $f\#\boldsymbol{\beta}=\boldsymbol{\alpha}$ that maps a pattern f and the analysis state β into the analysis state ∝ . Then we introduce the following operations on the patterns.

The catenation fg of patterns f and g is the function defined by the conditional expression $\mathrm{fg\#\alpha = p(f,\alpha) \rightarrow g\#(f\#\alpha)}$. The alternation f∨g of patterns f and g is the function defined by the conditional expression $\mathrm{f{\vee}g\#\alpha = p(f,\alpha) \rightarrow (f\#\alpha);\ p(g,\alpha) \rightarrow (g\#\alpha)}$. The iteration $\mathrm{f^*}$ of a pattern f is the function defined by the conditional expression with the infinite number of members $\mathrm{f*\#\alpha = \neg p(f_1,\alpha) \rightarrow \alpha;\ \neg P(f_2,\alpha) \rightarrow f_1\#\alpha; \ldots ; \neg P(f_i,\alpha) \rightarrow f_{i-1}\#\alpha; \ldots}$. The n-th power of a pattern $\mathrm{f^n}$ is the function defined by the recursion $\mathrm{f^1\#\alpha = f\#\alpha}$, $\mathrm{f^n\#\alpha = f\#(f^{n-1}\#\alpha)}$.

The introduced functions form the pattern algebra the elements of which in contrast to Kleene algebra are the analysis states, i.e. pairs $(\mathrm{v,i})$, where $\mathrm{v}\in\Sigma^*$ and i is an integer. The nontrivial constants of the introduced algebra are primitive patterns recognizing the occurrence of words consisting of the only character of the alphabet Σ. The constant 1 of this algebra presents the identity pattern. It has the following properties: if f is a pattern, then $1^*=1$, $\mathrm{1f=f}$, $\mathrm{f1=f}$, $1\vee\mathrm{f}=1$, $\mathrm{f}\vee 1=\mathrm{g}$. The pattern g is the everywhere defined pattern if $\mathrm{g\#\alpha = p(f,\alpha) \rightarrow (f\#\alpha);\ \neg p(f,\alpha) \rightarrow 1}$.

The regular expression language agrees with the set of formulas of the pattern algebra. The operation of applying pattern is similar to the process of applying transition rules of an automaton, which was built by the regular expression. The difference is that a nondeterministic selection of transition rules of the automaton is replaced by the arranged invocation of checks in the expression of alternation operations. By this feature the pattern algebra is deterministic semantics of the language of regular expressions.

The analysis states α and $\beta = f(\alpha)$represents the word v in the form of the concatenation axc, i.e. the pattern f recognizes the word x as a part of v. The set of all words that are recognized by the pattern f forms the language $L[f]$ recognizable by the pattern f. It is easy to verify that $L[f] \subset L(f)$.

The complement operation is included often in the signature of the Kleene algebra for the convenience of practical use. If a and b are regular expressions then the complement $(a-b)$ define s the language $L(a-b)$ of words in $L(a)$ but not in $L(b)$. The negation operation is more convenient in pattern algebra, it is defined as $\neg\#\boldsymbol{\alpha} = \neg p(f, \boldsymbol{\alpha}) \rightarrow \boldsymbol{\alpha}$.

The language substitution

The language substitution is a rule of operating strings of symbols. It is an extension of the word substitution rule of Markov algorithm. The language substitution can be specified as a triple (L_1, L_2, ϕ) where L_1, L_2 are languages and the function $\phi: L_1 \rightarrow L_2$ [Cheremisinov, 1981]. The apply function of language substitution $\phi: L_1 \rightarrow L_2$ to input string produces the word W from a word V (i.e. input string) in the following way:

1. Check to see whether any of $x \in L_1$ can be found in the word V.
2. If none is found, the apply function is undefined.
3. If one $x \in L_1$ is found to form word W the first of them can be used to replace the leftmost occurrence of the matched text in the input string with $y = \phi(x) \in L_2$.

A pattern f that recognizes language $L[f]$ with juxtaposed single-valued function $\phi: L[f] \rightarrow L_2$ is a pattern with side effect [Sebesta, 2009]. The operation $\beta = f(\alpha)$ represents the basic effect of the pattern. The side effect of a pattern can be another pattern. The patterns with side effects are the types of recursive functions. As a class of general recursive function coincides with the class of Turing computable functions (Turing–Church thesis), so patterns with side effects are the effectively calculable functions. The class of patterns with side effects is Turing complete or computationally universal. The consequence of the algorithmic completeness of the regular expression patterns with side effects results in the possibility of recognition of any type of language in Chomsky's classification, not just the class of regular languages.

Algebra of patterns with a side effect is built by the modification of the interpretation of the constant 1 of this algebra, i.e. the identity pattern. In this algebra only the identity patterns has side effects. Algebra of patterns with a side effect contains the set of identity patterns, which differ by their side effects. An identity pattern with a side effect corresponds to a function that is applicable to the analysis state when the previous pattern was applicable too. If the side effect defines the substitution of a word that was recognized by the previous pattern, then the catenation of the identity pattern with the side effect corresponds to a word substitution rule of Markov algorithm. The regular expressions of the algebra of patterns with a side effect define Markov algorithms. The application of this regular expression to the word V is the substitution of languages. Thus, regular expression of the algebra of patterns with a side effect is the algorithm for transforming words by substitution rule based on the constituents set of the regular expression. The construction of this algorithm is implemented by "embedding" the identity pattern with the side effect into the analysis procedure. Operations representing the side effects of identity patterns specify the algorithmic basis of language substitution functions.

Regular expression with the semantics of the pattern algebra could be imagined as a program on a special programming language. Application operations of the primitive patterns define the set of operations that make up this language. The control operations of the programming language correspond to the functional forms of regular expression. They specify the means of sequencing the application of primitive patterns. The variables whose values represent parsing states are not explicitly referred to the program text, their existence is assumed for each such program.

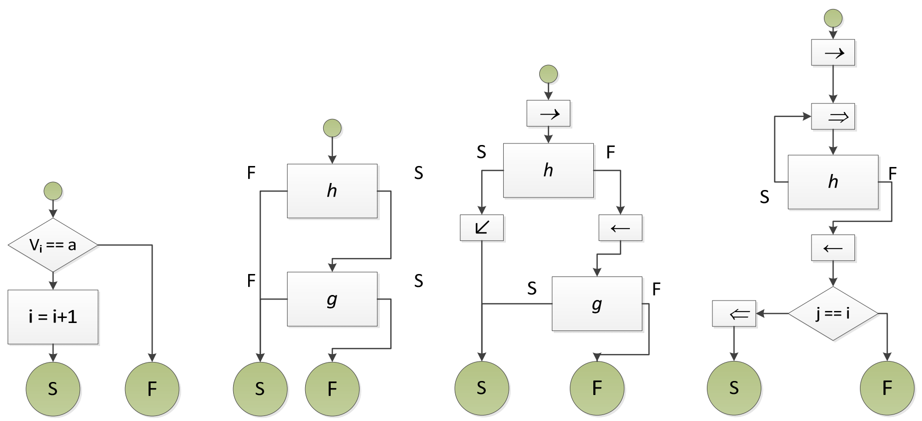
The prototypes of the algebra of regular expressions with a side effect are macro languages. In these languages a macroinstruction is a rule or pattern that specifies how a certain input sequence (macro-call) should be mapped to a replacement output sequence according to a defined procedure (macro-procedure). The mapping process that instantiates (transforms) a macro use into a specific sequence is known as macro expansion. Language substitution (L_1, L_2, ϕ) in the form of macro is described by a pattern describing characteristics of language L_1 and by algorithm of constructing the language L_2. In the case of language substitution in the form of regular expression, the constituents of a regular expression are defined as macro-call formal parameters; words that are recognized by constituents are the actual parameters; the side effect forms macro-procedure. The regular expression language is similar to XSLT language, but unlike XSLT, the regular expression language is better suited to handle unformatted text.

The compilation of regular expressions with a side effect

Let U be a set of variables and V be a set of possible values of variables of U. Let us define a machine Ω which memory states are functions from U into V, so the set S of all memory states coincides with the set of functions $s: U \to V$. The control state of the machine Ω is a pair (v,i) where v is a word and a i – an integer. $v \in L(\Omega)$ defines the text of the program for the machine Ω, $v \in \Sigma^*$. Let the programming language $L(\Omega)$ is a regular expression language. The set $\Psi(v)$ of control states of program v of the machine Ω coincides with a set of analysis states of program v. The set $\Psi(\Omega)$ is the union of $\Psi(v)$ for all programs of the language $L(\Omega)$.The instruction set of Ω is a set of patterns with side effects defined on Cartesian product $\Psi(\Omega) * L(\Omega)$. If the pattern f is an instruction of Ω, then the set of analysis states coincides with the set of control states where the instruction is applicable. The side effect of f is the function that transforms the given memory state into resulting one. The machine Ω is specified by the pattern $(f_1 \vee f_2 \vee ... \vee f_n)*$, where f_i is a member of the instruction set of the machine Ω. The regular expressions with side effect are programs of $L(\Omega)$.

Given the program p of the machine Ω. For each control state $\alpha \in \Omega(p)$ we can describe a transition function $\pi_i : S \to \Omega(p)$, that defines control state of the machine Ω after execution of an instruction allowable in control state α. The union of all π_i specifies a function $\pi : \Omega(p) * S \to \Omega(p)$ that is denoted as a schema of p. The schema of p can be represented by a graph $G(\Omega(p), \pi)$. The vertices of G correspond to the control states and the edges are marked with values of memory variables from U. A program scheme describes (models) a concrete program. Concrete programs can be obtained from schemata by means of interpretation which consists in bringing some concrete variables and operations into correspondence with formal variables and operations.

An instruction of the machine Ω that changes only the state of the memory is called a converter, the command that does not alter any memory state is called a resolver. Resolvers are patterns without side effect. In graphical scheme representation of recognizers are denoted by diamonds, converters – by rectangles (Figure 1).



a) b) c) d)

Figure 1. The templates of program schemata which perform analysis of character strings

Memory U for the machine Ω performing analysis of character strings is composed of the variables u,i representing the analysis state of the parse string; stack to store numeric values and work variable j of integer type. The converters with labels $\rightarrow$, and $\leftarrow$ $\swarrow$ define the operations that store he variable i on the stack, which is empty at the beginning; load the variable i from the stack; delete the top-most element of the stack, respectively. The converters marked by $\Leftarrow$ or $\Rightarrow$ define the operations that push the work variable j into stack or move j into i.

The program flow of program scheme p in the memory state s_0 is defined as follows. The program flow is the travel under the scheme step by step. A program scheme has always one active instruction, which is pointed by a control state. The program flow starts from the control state $(p,0)$ in a state s_0 of the memory U of the machine Ω. The execution of a converter y is the transition of memory state x into state $y(x)$. The execution of a recognizer does not change the current memory state but selects one of the outgoing arcs of the resolver to continue travel under the scheme. If recognizer is labeled by a variable u, then we select the arc labeled by the value $u(x)$. Execution of the program scheme is completed when traveling lead us to the scheme output. The current memory state is a result of execution of the program scheme in this case. Otherwise, the result is undefined. Thus, we decided to consider program scheme as an imperative language and to use a structural operational semantics. The operational semantic system in its entirety is an interpreter that links the program text to the set of possible executions.

For the synthesis of the program scheme that implements the algorithm specified by the pattern a we use reverse Polish notation of a pattern R. The synthesis algorithm uses a stack of a program schemes. We parse reverse Polish notation of the pattern R. If current token is a primitive pattern, which recognizes a single symbol, then we push into the stack the program scheme on Figure 1,a. If current token is an operation of a regular expression, then we form new program scheme from the popped one and push it into the stack. To form a new program scheme we use stereotype of the program scheme on Figure 1,b (1,c,1,d) if an operation is the catenation (alternation, iteration). As a result the stack contains the only scheme which is a program scheme that implements this pattern R, if the initial regular expression is syntactically correct.

The interpretation of converters and recognizers of program schemes as expressions of conventional programming language is the basis of the compiler of regular expressions, which performs the analysis of character strings. Selection of the object language is mainly determined by the capabilities of the programming system in which the language is used as an input. Object language should allow the use of character string data type, and where there is the possibility to access the individual characters in a string for this string data type.

The regular expression language and a compiler form a programming system of patterns which performs analysis of character strings. Currently the compiler of this system is a preprocessor that converts patterns into programs in the programming language C. To represent the analysis state of the character string the pointer $char * A$ is used in programs in C. The stereotype of primitive patterns (Figure 1,a) is given by $\text{int B} = *(\text{A}++) == \text{n};$ where n is a recognizable symbol. The catenation (Figure 1,b) is the stereotype $h\, if(!\,B)\{\, g\,\}$. The alternation (Figure 1,c) is the stereotype $\text{P}.push\_back\ (\text{A});\ h\, if\ (\text{B})\{\ \text{A}\ =\ P.pop\_back\ ();\ g\ \}\, else\, \{\ \text{P}.pop\_back\ ();\ \text{B}\ =\ 1;\}$. The iteration (Figure 1,d) is the stereotype

$do\,\{\ \text{P}.push\_back\ (\text{A});\ if\ (!*B)\{\text{B}\ = 0; break;\}\, h\ \text{B}\ = (!\,\text{B});\}while\ (!\,\text{B}\,);\ if\ (!\,\text{B}\,)\text{B}\ = -1; else\ \{\text{A}\ = \text{P}.pop\_back(); \text{B}\ = 0;\}$■

Analyzers based on deterministic finite automaton have linear time complexity $O(|S|)$ for strings of length $|S|$, because it does not need to be rolled back (do not check twice a symbol of the analyzed text). The analyzers, which are built by programming system of patterns, have the rollbacks as it is seen from the stereotypes on Figure 1.

Conclusion

We have presented an operational semantics for the regular expression language. Our semantics of regular expression gives the basis for building programming tools for the language analysis. The pattern algebra represents both functional model of the constituent analysis of texts and the control flow model of primitive templates execution during text analysis. Generally accepted formalism to describe the structure of the constituents of the text is the grammar. The use of patterns with a side effect is able to make the universal formalism from the regular expressions, as well as grammar. The implementation of string analysis based on regular expression patterns with a side effect is potentially more effective than the grammar parsing algorithm, because it is possible to control the sequence of applications of grammar rules. The grammar parsing algorithm is not a part of the grammar formalism. Using grammar rules as patterns is more natural in the case of a language substitution, especially in simple cases, compared with the attribute grammars, because to compute language substitution $f\ (x)\ =\ y$ by the attribute grammars we have to take into account the parsing algorithm. However, the concern about the control of the sequence of grammar rules is not so good. The optional degree of freedom may increase the complexity of the description analysis.

The programming system of regular expression patterns has been successfully used for the construction of the analyzers of interchange data formats [Cheremisinov, 2013].

Bibliography

[Friedl, 2006] J.E.F.Friedl. Mastering Regular Expressions, 3rd Edition, O'Reilly, Sebastopol, CA, 2006.

[HMU, 2000] J.Hopcroft, R.Motwani, and J.D.Ullman. Introduction to Automata Theory, Languages and Computation. Addison Wesley, 2000.

[Kozen, 1997] Dexter Kozen. Automata and Computability. Undergrad. Texts in Computer Science. Springer-Verlag, 1997.

[Kleene,1956] S.C.Kleene. (1956), Representation of events in nerve nets and finite automata, in Claude Shannon & John McCarthy, ed., 'Automata Studies' , Princeton University Press, Princeton, NJ , 1956, pp. 3-41 .

[Aho, 1990] Aho A. Algorithms for finding patterns in strings // Handbook for theoretical computer science, MIT Press. – Vol. A.,1990. – P. 257-300.

[Boost, 2002] Jeremy Siek, Lie-Quan Lee and Andrew Lumsdaine. *The Boost Graph Library*. Addison-Wesley, 2002.

[Thurston, 2007] A. D.Thurston. Ragel State Machine Compiler User Guide http://www.colm.net/files/ragel/ragel-guide-6.9.pdf, 2007.

[McCarthy, 1960] J.McCarthy. Recursive function of symbolic expressions and their computation by machine, part 1. Comm. ACM – 1960, v.3, n. 4. – P. 184-195.

[Cheremisinov, 1981] D.I.Cheremisinov. Language substitution programming // Programming, Moskva: Nauka, – 1981. – № 5. – P. 30-37.

[Sebesta, 2009] R.W.Sebesta. Concepts of Programming Languages, 9th Edition, Addison-Wesley, 2009.

[Cheremisinov, 2013] D.I.Cheremisinov. "Design automation tool to generate EDIF and VHDL descriptions of circuit by extraction of FPGA configuration,"*Design & Test Symposium, 2013 East-West*, Rostov-on-Don, 2013, pp. 1-4.

Authors' Information

***Dmitry Cheremisinov** – The United Institute of Informatics Problems of National Academy of Sciences of Belarus, leading researcher, Surganov str., 6, Minsk, 220012, Belarus; e-mail: cher@newman.bas-net.by*

Major Fields of Scientific Research: Logic design automation, System programing

ANALYSING THE REQUIREMENTS OF MODERN BUSINESS TOWARDS USER INTERFACES

Margarita Atanasova, Anna Malinova, Hristo Krushkov

***Abstract**: A detailed and accurate business requirements analysis is critical to the success of any project. In order to maintain or increase their competitiveness software companies strive to deliver good user experience and intuitive user interfaces, which are effective, usable, responsive, visually appealing and properly tested. The research presented in this article is based on our analysis on the user interfaces of large amount of modern business information systems and our work with several big and forward-looking companies. The article summarizes their business requirements towards user interfaces, describes the implementation of the corporate identity into the designing process and presents the specifics about visualization of reports. We research how clients' requirements on the responsiveness of the design depend on the country and region, where the product is going to be distributed. It is also explained the importance of properly conducted usability testing and how to achieve optimum results depending on the number of participants. It is also represented how intelligently selected colour palette, shadows and shapes help the intuitive handling of the system and easier perception of information. This article describes generating and managing reports as main functionalities in business information systems. We examine different types of report customization, the use of visualization generator to create and publish interactive reports, applying different themes and dynamically altering the display density of table reports. In this article we also discuss different sample design solutions.*

***Keywords**: user interfaces; business requirements; user experience; reporting; usability testing*

***ACM Classification Keywords:** D.2.2 Design Tools and Techniques; D.2.1 Requirements/Specifications*

Introduction

In the last decades, forward-looking companies like Apple, Google and Microsoft focus on the design of exceptional products and delivering outstanding user experience. Their aim is to help the user feels comfortable with the use of the system. That way the company builds its relationship with the customers, earns their trust and builds on its corporate identity. Inspired by this approach, other companies in the sector began to strive to deliver good user experience and intuitive user interface in order to maintain or increase their ability to compete on the market.

Designing user interface is often not an easy task. The team have to start creating wireframes and mockups of the design once enough technical requirements are gathered, systemized and analysed. A successful user interface focuses on users and their day-to-day tasks. The participation of clients and customers in the designing process is proven to be an effective practice [Trischler, Kelly, 2016]. In the user-centric approach user interfaces are driven by user profiles and user requirements.

The business analysis team and the user experience team should identify the users and user roles of the system. This is done through various techniques such as interviews with consumers, clients and empirical observations. The identification of valuable users or those who will contribute to the project is also seen in other areas such as the development of innovations [Stockstrom, Goduscheit, Lüthje, Jørgensen, 2016].

Crucial part of any software functionality is also reporting. Generating and managing reports is a requirement of any modern business and the information the end user receives from it must be customizable through themes and visualization generators in order to achieve flexibility and to be easily adjustable into the user’s daily workflow.

This article presents a detailed analysis on the requirements of modern business towards user interfaces. We examine the need of effective, usable, responsive and visually appealing user interfaces. We research how clients’ requirements on the responsiveness of the design depend on the country and region, where the product will be distributed. It is also explained the importance of properly conducted usability testing and how to achieve optimum results depending on the number of participants. This section also represents how intelligently selected colour palette, shadows and shapes help the intuitive handling of the system and easier perception of information. At the end of the section we research other requirements like the implementation of the corporate identity guidelines and the need of visualising Big Data on mobile devices.

This research describes generating and managing reports as main functionalities in business information systems. We examine different types of report customization – the use of visualization generator to create and publish interactive reports, applying different themes and dynamically altering the display density of table reports.

The Requirements of Modern Business towards User Interfaces

The research presented in this article is based on our analysis on the user interfaces of large amount of modern business information systems and our work with several big and forward-looking companies.

The first company is Australia's leading provider of mobile phones, mobile devices, home phones and broadband internet. The second company is a digital infrastructure management company, which provides the only monitoring platform, engineered for speed at scale and which seamlessly integrates infrastructure performance and end-user experience data. The third one is a multinational technology company that designs, develops, and sells consumer electronics, computer software, and online services. We summarized their business requirements to user interfaces into several sections:

- Effectiveness
- Usability
- Responsiveness
- Usability testing
- Visually appealing
- Other

Effectiveness

Effective is work, action or process that produces very good outcomes. The effectiveness of a single user interface helps the users to perform quickly and efficiently their daily tasks. For that purpose, before building a prototype, the tasks that users perform on the system in their daily work should be analysed. The term "task analysis" incorporates the application of various techniques for identifying and understanding the structure, sequence of actions and parameters of a task [Crystal, Ellington, 2004]. According to [Guerrero, Lemaigre, Gonzalez, Vanderdonckt, 2008] the task model is a logical description of the sequence of actions that are intended to help the user to reach any destination in an interactive system. Basically, the method of tasks analysis incorporates three main elements [Oliveira, Lepreux, Lepreux, Kolski, Seffah, 2014]:

- The iterative approach, which performs a sequence of actions described in the models;
- Models that capture aspects of a problem and convert them into specifications;
- Software tools used to accelerate and facilitate the process.

Task analysis are used to identify:

- What are the users trying to achieve? What are their goals?
- What actions they have to perform in order to complete their goals?
- What type of experience helps them complete those tasks: personal, social, cultural?
- How does the environment influence their actions?

There are several types of task analysis. The most popular of them are:

- Hierarchical Task Analysis (HTA) [Promann, Zhang, 2013]. In this approach, complex tasks are decomposed into simpler;
- Cognitive Task Analysis [Gallagher, Prestwich, 2013]. The purpose of this approach is to analyse the tasks that require cognitive activity from the user such as problem-solving, decision-making, attention and judgement, memory [Crandall, Klein, Hoffman, 2006].

The task analysis approach is often used in the earliest design phase before the prototyping. It can be used to support other aspects of creating better user experience:

- Gathering requirements;
- Creating a content strategy and information structure of the system (web page);
- Wire framing and prototyping;
- Usability test sessions.

Task analysis are used to decompose the main user tasks into simpler ones. That way the workflow of the use cases is identified. Knowing the workflow, the wireframe designers can produce accurate mockups that will lead the user through the interface effectively.

Usability

In the prototyping phase implementing the principles of the good design is essential in order to create effective and intuitive user interface [McRee, Anderson, Wilson, 2010]. The usability of the system could be measured by test sessions and heuristic analysis [Rubin, 2008]. Independently from the target group, most information systems will always have two types of users: novice and experts. Novice users should be able to learn quickly how to use the system, while the experts should be able to perform their everyday tasks efficiently. This may even mean the development of various interfaces. But in most cases the designers manage to combine the requirements of different types of users in a single interface.

A good user experience and interaction design technique is progressive disclosure. This approach sequences information and actions across several screens thereby the user stays focused on completing a task and does not feels overwhelmed. For example, this is a good approach for registration forms or where filling large amount of data is needed. It is also used for setting up a software or the so called “start-up” wizards.

Example: The user needs to perform several tasks in order to start monitoring their network, e.g. fill in many fields in different categories (for example IP address range or Subnet block, type of device, device name, objects and indicators, a time span filter, etc). Single and complex user interface might produce clutter, confuse the user and lead to uncomplete task or giving up. Disclosing information step-by-step and showing only the essentials for the current step helps the user manage complex and feature-rich applications. Therefore, we divided "Enter information" task into five-steps wizard with form validation as shown on "Fig. 1". Hereby even novice users managed to start monitoring their devices.

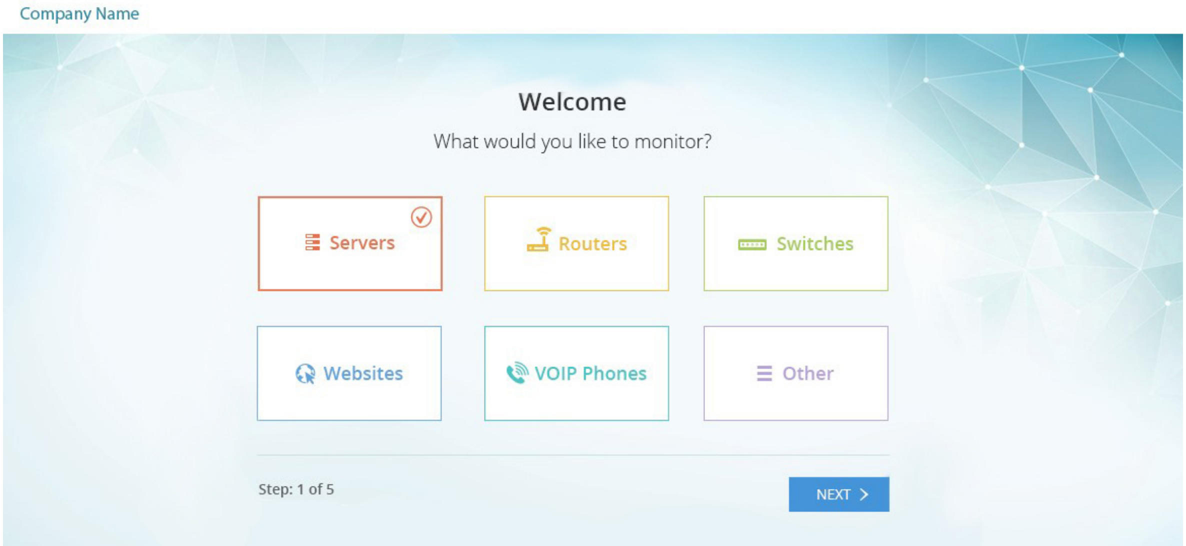


Fig.1. Progressive Disclosure helps users handle complex tasks.

Responsiveness

Interface that adapts to the device on which it is displayed is called responsive [Marcotte, 2011]. There are several modern techniques used for creating responsive UI: media queries (CSS3 module which detects the specifics of the screen), fluid grids (columns that make up the adaptive mode), flexible media files and images. In the last decades this is actually one of the main demands from the business stakeholders - the UI has to look good on all types of devices - from mobile phones with different screen sizes and resolutions to big computer screens and smart TV-s. Testing on all types of devices in the development process is required, but it is achievable as there are multiple types of good browser emulators representing the performance on the real physical devices.

.Clients' requirements on the responsiveness of the design depend also on the country and region, where the product will be distributed. For example, stakeholders from China, require the mobile version of the application to look properly on the most used mobile OS platform for Asia. According gs.statcounter.com this is Android with nearly 71% of market share for the period (Jan 2016 - May 2016) [gs.statcounter.com – a), 2016]. In the United States of Amerika iOS leads with 58%, followed by Android with 39% for the same period of time. That is why the clients from USA demand visually appealing user interface mainly on the both operating systems. Windows Phone is used only by 0,92% of the mobile users for the same period in USA [gs.statcounter.com – b), 2016]. In some countries in Europe the market share for Android is even higher than in Asia. For example, in Bulgaria nearly 81% of the mobile users have Android smartphone, 14% - iOS and only 3% Windows Phone [gs.statcounter.com – c), 2016]. As a whole, in Europe most of the users (58%) use Android OS, followed by iOS with 36%. In Australia though iOS has a leading role with 67% vs 30% for Android [gs.statcounter.com – d), 2016].

Usability Testing

After having developed the first prototype, UX designers test their "assumptions" for efficient and intuitive design with a targeted group of users. To test the quality and level of intuitiveness of the user interface there can be used a variety of tools and frameworks [Islam, Bouwman, 2016]. Usually the test sessions with users are recorded and later on used for further analysis. After the test session issues are identified, fixed and if there is enough budget, the company performs a second usability testing.

User interfaces should provide their users the most efficient, easiest and fastest way of completing a task. The problem is that the users are different and the best way for completing a task for one might not be the optimum way for other. That is why the system should provide at least several approaches for the users to accomplish their everyday tasks. Implementing different ways of completing a task usually improves the usability palpably.

User interfaces should provide their users the most efficient, easiest and fastest way of completing a task. The problem is that the users are different and the best way for completing a task for one might not be the optimum way for other. That is why the system should provide at least several approaches for the users to accomplish their everyday tasks. We analyzed the users’ needs and feedback from usability test sessions on a sample prototype of Big Data monitoring system conducted in a live environment. The “shadowing” method of observation and user’s thinking out loud when going through the test case scenarios led us to several conclusions. While some of the tasks were completed with ease by part of users, others experienced not only difficulties but even inability to move forward. We analysed the work experience, personal user habits and preferences and how they influence the user decisions and expectations. Basic example is renaming a tab. Users that are used to Microsoft products click twice on a HTML tab element and they expect the inside text to transform into a text field. The other users use right mouse click in order to open an additional menu with options like “Delete”, “Rename”, etc. This resulted in developing a sample adaptive prototype of a system (“Fig. 2”), which offers different ways of completing a task. The second usability test session with different users, both novice and experts, showed task completion by 92%, which is 22% higher than the first test session.

Optimal Results through Number of Participants

Prototypes are useless unless they are used for testing with real users. The obtained information is used for the next iteration of sketching, prototyping and testing. Usually software development is associated with a limited budget and time. Therefore, testing at an early stage prevents some of the errors at a later stage. According to a mathematical model there is no need to test with large amount of users [Nielsen, Landauer, 1993]. The best results which take into account time, budget, and number of issues found are obtained by testing of no more than five users. The author of the mathematical model recommends budget to be allocated between three different test sessions with five users each for optimal results.

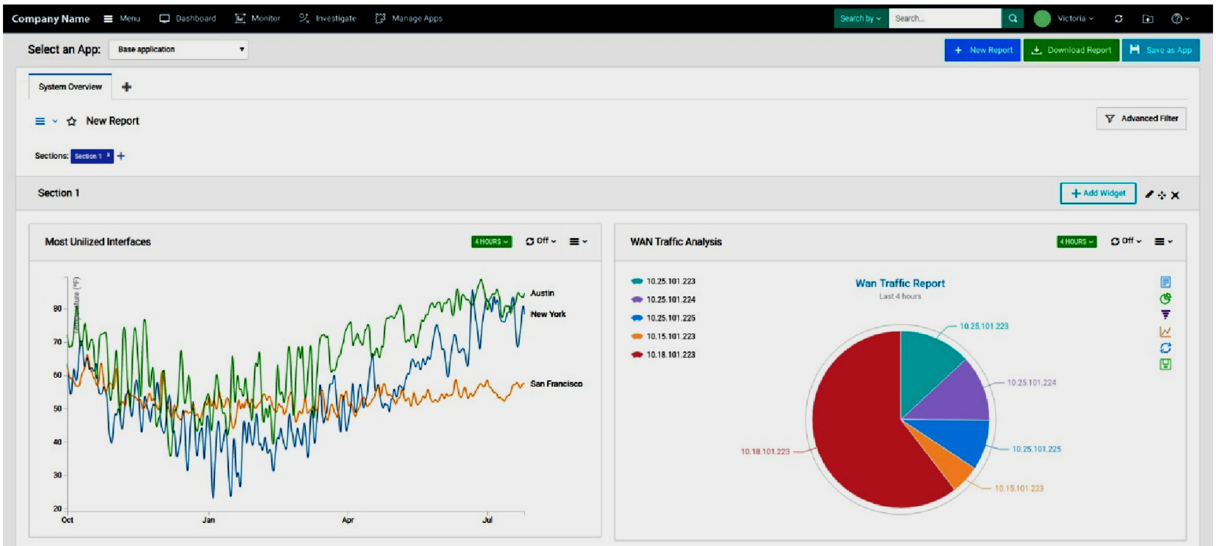


Fig. 2. Sample prototype of a system that implements different approaches for completing a task

Personas

If for some reason the GUI cannot be designed and tested with real users, the designer can use custom prototypes or personas [Nagel, 2016]. “Persona” is a descriptive model of a certain type of user based on data from a customer survey [Marshall, Cook, Mitchell, Summerskill, 2015]. Building the software personas requires describing their profession, habits, goals, motivation and daily tasks, which they will perform on the system. All this Information can be collected through business analysis.

Visually Appealing

Business Needs

Almost everyone could tell whether if they like or not a user interface. The combination of clean and minimalist design, intelligently selected colour palette, light shadows that show the depth of objects helps the intuitive handling of the system and easier perception of information. And this is exactly what the modern business is looking for. On “Fig. 3” is shown an authors’ user interface of a data monitoring system, that is created according clients’ requirements of modern, well-structured and visually applealing design. All of the selected colours are used to guide users’ attention to the different sections. The main background image in the center part of the page is used to relax the user and make them feel comfortable in completing their day-to-day tasks. This image is dynamically changing according the seasons, day time and user’s visits so that every time the user opens the system is surprised with something new. This is also part of the system-user interaction.

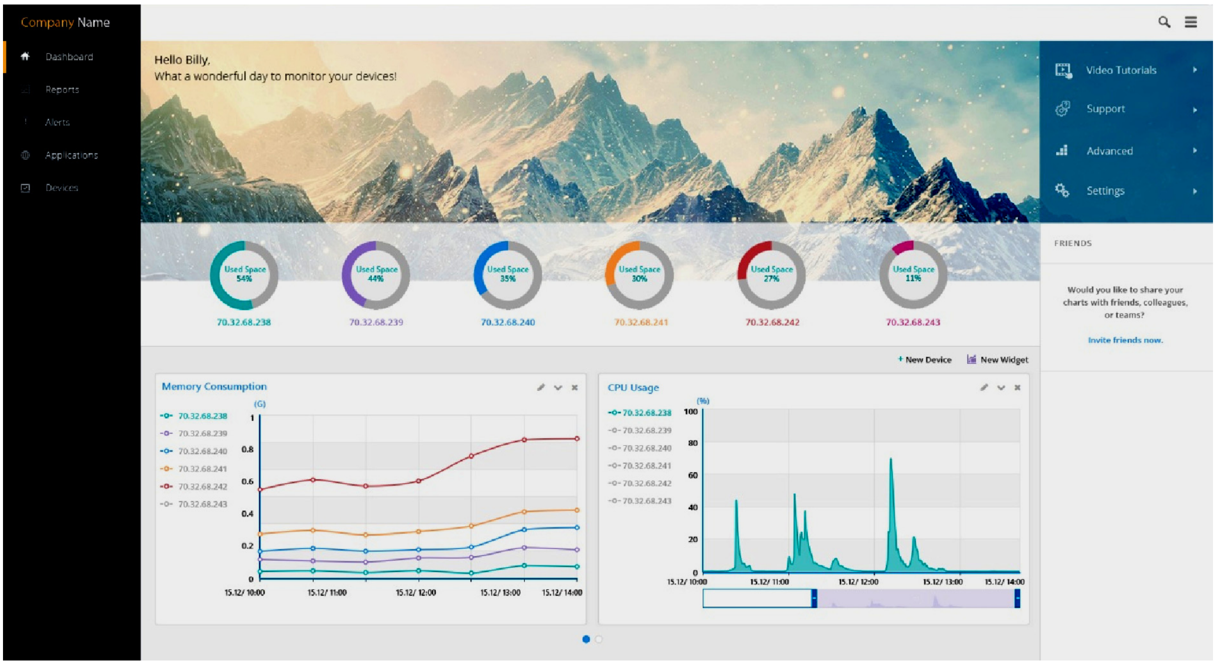


Fig. 3. Modern and minimalistic, visually applealing example of a user interface.

Just like in fashion, the sense of beauty in the graphical user interfaces evolves and changes in time. While the gradients and 3D like buttons were considered pretty in the past, now user interfaces strive for clarity and simplicity. There are many factors that influenced the simplification of the design.

Design Trends and Movements

- ***Minimalism***

The Minimalism arose in the 1950s [Encyclopedia Britannica, 2016]. The main principle of this movement is simplicity of shapes and content.

Minimalism uses intelligent and well-planned colour palettes. Fonts are clean and easy to read. The images are small in number but have strong impact on the viewer.

- ***Flat design***

With the release of Windows 8 in 2011, Microsoft starts the trend of the "Flat design". This is type of minimalistic design, which emphasizes on the usability and simplicity. There are no complicated graphics, heavy colour combinations but rather simple shapes and clean tones. All 3D elements are removed and there are no more shadows, gradient or textures. This approach points out several advantages: First the load time is minimized (the cascade style sheets have smaller size, most of the backgrounds are CSS based instead of images). Second the use of fluid grids accelerates the development process as that way the application could be easily tailored for mobile devices. Designers do not use any heavier shadows, textures and fixed images as they will not be visually appealing on mobile phones. The flat design trend helps users perceive the content faster and stay focused while completing a task. The similar look and feel of many modern software products on one hand does not surprise the users and makes them feel comfortable but on the other hand more and more web systems start to look tedious, uninspired and uniform.

- ***Skeuomorphism***

While Microsoft works on their flat design, Apple introduces new approach called Skeuomorphism. It uses graphics resembling objects and materials from real life - wood, leather, paper. But in 2013 Apple abandon this approach and adopts the minimalistic and flat vision for iOS7.

Most likely "Flat" design will also be replaced by something new and exciting as it happened with the style before that. The usability testing shows that this approach has drawbacks such as lack of visual hint that the element can be clicked.

- ***Material Design***

In 2014 Google introduced Material Design which began to gain popularity in 2015 after YouTube, Gmail and all other mobile and desktop Google applications adopted it. It combines classical principles of good design, science and new technologies of today. This visual language is based on how people perceive real objects. With the help of surfaces, shapes, angles, shadows and shades, animations and images the designers draw the attention of the users and hint the different functions of the elements [Google material design, 2016].

Other

Corporate Identity

Every company has its own corporate identity. In general, it consists of a logo, corporate title and set of guidelines. These guidelines describe how the identity is applied and include approved colour palettes, typefaces, page layouts and other such. Every customer demands the user interface of their product to be created according to the company’s corporate identity. This requires designer’s attention to several important elements:

- ***Logo***

The logo should not be changed. It might only be resized according to the space given in the header. Typically, a good logo should have black and white versions as well as colourful. If the header background is white the designers typically use the colourful version of the logo. If the colour is black, they might use white or colourful version of the logo. The most appropriate place for the logo in the user interface is the top left corner. It is often used for “Home” button as well. According to the Gutenberg Diagram in web design this is the Primary optical area [Lidwell, Holden, Butler, 2010].

- ***Colour palette***

Colours should be used according to the corporate identity. Typically, every company has selected colour palette which communicates with the clients in the best way. For example, Telstra, Australia's leading provider of mobile phones, mobile devices, home phones and broadband internet, chose an identity that incorporates the full spectrum of colours, tones and emotions after their rebranding in 2012. “That way Telstra can communicate with greater meaning and with the flexibility to talk to different customers at different times, in a manner that’s not only relevant but also engaging” - says Joao Peres, the designer behind the new Telstra identity [Telstra identity, 2016].

- ***Colours in the charts***

Creating a user interface of an information system with charting functionality requires careful planning of the colour palette that is used by the graphing engine. If the customer has more colours (like Telstra), it is easy to create beautiful and communicating colour palette for the charts. But if the company has only two primary colours it is nearly impossible to build a good charting colour palette using only shades of these two colours. That is why most companies which do not have enough colours in their corporate identity pack do not require using their colours in the charting engine. Instead they use all colours in the rainbow so that the charts represent the real data using the meaning of the different colours.

- ***Typeface***

Implementing font family of the brand is one of the client’s important requirements during the designing process because the typeface is also used to communicate visually with the users. One of the most used font families for web is Open Sans, developed and used by Google [Google fonts, 2016]. It can be integrated into a web system for free and it is popular in flat design-style web design. Similar to Microsoft’s Verdana it is extremely well readable on all devices and screens. It is also used as a default typeface in the Mozilla style guide [Mozilla Style Guide, 2016].

After careful exploration of the company's identity and incorporating all brand guidelines, the user interface corresponds better with the users, builds trust and good relationship as it becomes part of the software family of the company as shown on "Fig. 4".

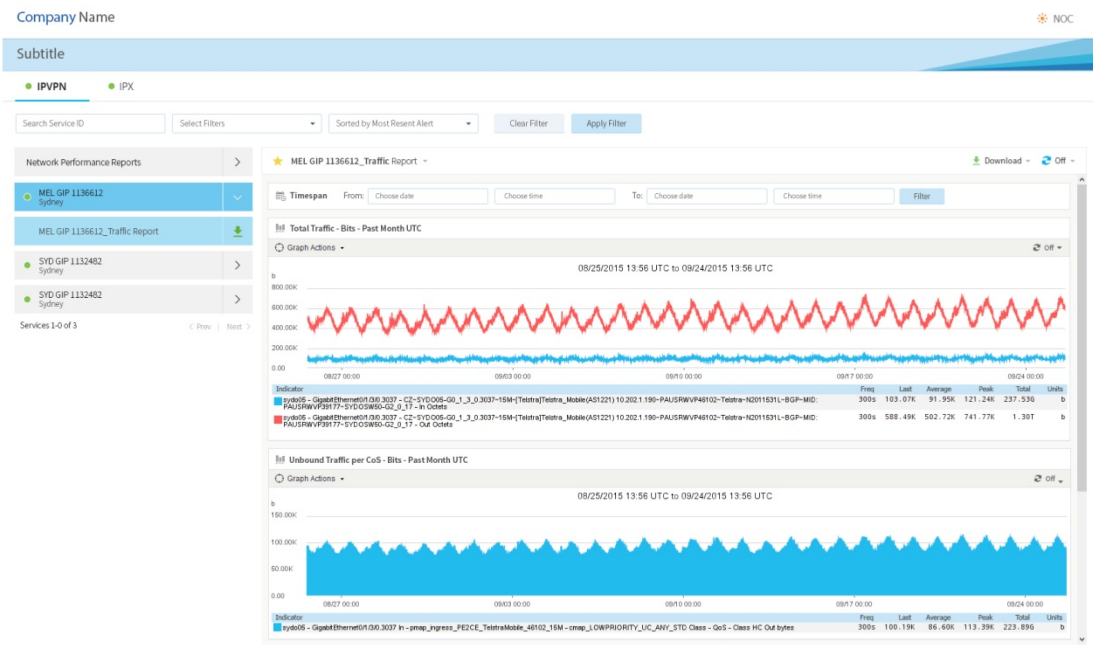


Fig. 4. User interface designed and build according company's corporate identity guidelines.

Visualizing Big Data on Mobile Devices

Typically, most of the clients require a responsive user interface. But they are not aware of the difficulties that working with Big Data introduces when there are tens of millions of data to present. Authors research different approaches and techniques for collecting, analysing, processing and visualization of Big Data on mobile devices by using cloud computing and NoSQL database management systems like Google's Big Table, Apache's Cassandra, MongoDB, HBase, DynamoDB, Voldemort and others [Wang, Liu, Soyota, 2014].

With mobile-cloud computing user can access Big Data anywhere at any time. If the software company that provides the Big Data analytics software does not have integrated cloud computing solution, the mobile version of the software will work slower and the loading time will be much bigger. In the last decades the mobile devices became more powerful. Many of the desktop operations could be done using a smartphone. Still, they are not powerful enough to handle analysing Big Data.

Example: The interface below ("Fig. 5") shows the mobile version of the software shown on "Fig. 4". The system generates very detailed reports showing hundreds of data points. There are two approaches on visualizing such data on small screens: The first one is to present a simplified graphic. It will show the basic flow of the data but it will not be accurate enough. The user will not be able to see the detailed data information. The second approach is to generate smaller version of the report and give the user the ability to zoom in and out on certain parts of it (as shown on "Fig. 5"). We adopted the second approach, based on the client requirement – the mobile version of the software should be the same as the desktop version. That means visually the report has to be identical on all screens and devices.

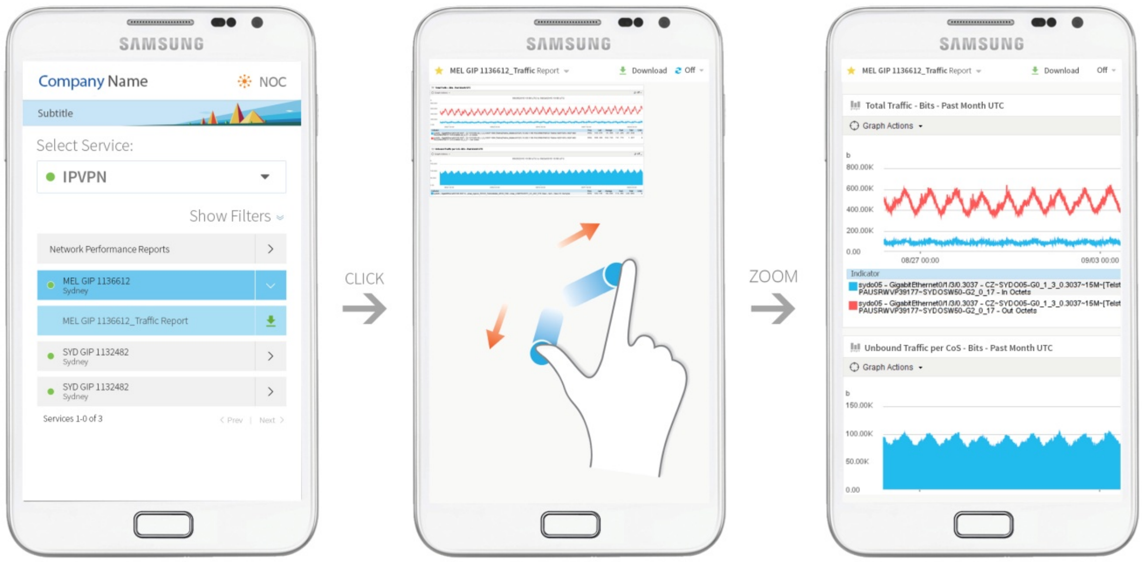


Fig. 5. Visualizing Big Data on mobile phone

Reporting

One of the main functionalities in a business information system is generating and managing reports. This component of the business software is the highlight of intel that the end user receives.

Basic Reporting Requirements for UI Controls

- ***Main UI Controls***

Typically, the users expect to have all basic operations implemented into a report system. That includes zoom in and out on the charts, drill-down (reaching to more detailed information by clicking on parts of the report - e.g. table row, part of a chart, etc.), change the time span and the filter, search by keywords, as well as the basic operations: create, view, edit, delete, plus duplicate and download/export report. Depending on the specific business types a report might have additional functions.

- ***Download, Export and Import***

Typically, a report could be downloaded/exported in several types: PDF, Excel, Microsoft Word, as an image file and other. More complicated reports, which will be imported into another system (e.g. Big Data analitics system), might be exported as csv or log files. The customer requirements depend on the use and storage space of the report. Microsoft Report Builder gives the users the ability to export reports as XML file with report data, CSV (comma delimited), TIFF file, PDF, MHTML (web archive), Excel, PowerPoint, Word. Users also want to be able to import different data sources into the data monitoring system.

- ***Print Report***

When previewing a report, the browser or the software application gives the user the ability to print out the document. According to studies [Liu, 2008], around 80% of the users prefer printing out electronic documents and do the reading from them. Using electronic source for reading helps you search keywords, but reading from printed document is much more user-friendly and less distracting. Participants in another study report that print documents enable easier comparisons to be made among all gathered resources [Liu, 2005]. It is very likely that users will continue printing out documents, therefore UI designers and front-end developers should focus on the print layout of the report as well.

- ***Chart Colours in Reports***

Colours for the charts should also be meaningful. The users get confused if a line in the line chart is red and if it’s meaning is good (e.g. the line does not represent a warning or an alert). Therefore, the colours should carry their own meaning. Many reports use traffic light system. Red for alert, orange/yellow for warning and green for normal. In addition, Big Data reports require more colours, e.g. more than 20, so the colour palette in these systems use bigger spectrum of colours. For example, the user wants to monitor 50 devices. For each one of the devices the report system draws a line in a different colour. If the list of the colour reaches its end, the colours starts from the beginning. Another approach is to generate colours with a computational algorithm.

- **Chart Types in Reports**

Another nice to have feature is dynamically changing the chart type. This is possible by adding the required chart types (e.g. line, pie, funnel, etc.) as small icons in the chart navigation. When the user clicks on one of the icons, the chart below immediately transforms (accompanied by a pleasant animation) using that type of draw. Currently this approach is implemented very successfully by Baidu, a Chinese IT company into their JavaScript library eCharts [echarts.baidu.com].

Customisable Charts and Tables in Reports

- ***Charts***

- *Visualisation generator*

Having a dashboard with similar diagrams and charts distracts users' attention and makes their everyday work slower. That is why creating a distinctive diagram for certain types of reports is significant. That could be achieved by using a visualization generator that can create and publish interactive charts. Typically, the goal workflow is as follows: First, the user selects a data source, e.g. Excel, xml, log, txt, etc. or device, device group, objects, indicators, etc. After that the user chooses a chart type, then applies a filter, e.g. time span, IP address range, indicator, etc.

- *Charting themes*

Interactive and customizable reports might be achieved by giving the user the ability to choose a theme. That way the creation of a report is transformed from everyday obligation into a game. Using the techniques of chromotherapy, the user spends some time playing with the colours which helps them relax and feel more comfortable. This experience is even more important for the users who have synesthesia [Cytowic, Eagleman, 2009]. Human brain and eyes detect even the smallest differences and nuances. Considering the time information technology specialists spend behind the screen that could have a significant impact.

A lot of chart properties could be customized and changed: background (type and image), title (text, colour, size, font weight, title position), legend (vertical, horizontal, visibility), tooltips (placement, background, colour, font size), animation (type, timing). Also there might be sample themes implemented with predefined different values of these properties, e.g. "Flat", "Monochrom", "Dark", "Light", etc.

- **Tables**

Tables in a report might also have different themes but their appearance is mainly limited to border type, colour, thickness, row colour, row mouse over effect, tooltips, title. Nice to have feature is altering the display density of the rows. Google Mail supports three types: Comfortable (37px height), Cozy (30px height) and Compact (25px height), where comfortable is the default row height. When the table has hundreds of records, one might want to change the view into compact so that more rows appear on a single page.

Conclusion

Looking at the next five years, the role of user interfaces will continually increase as companies compete to gain a larger market share worldwide. A weak implementation of the graphical user interface could lead to failure even a stable and well-designed software. Understanding the business needs in combination with deep and focused business requirements analysis will lead to the success any project. We divide business requirements into six different sections: effectiveness, usability, responsiveness, usability testing, visually appealing and other. In the section “Other” are included the need of accurate implementing the company’s corporate identity and visualizing Big Data on mobile devices. As additional business need for the user interfaces of business information systems we describe generating and customizing chart and table reports. In conclusion, three usability test sessions with no more than five participants each will eliminate most of the user experience issues which might occur in future.

Acknowledgment

This work is partially supported by the IT15-FMIIT-004 project of the Scientific Fund of the University of Plovdiv “Paisii Hilendarski”, Bulgaria.

Bibliography

[Trischler, Kelly, 2016] J. Trischler and K. Kelly, “User Involvement in the Design of New Library Services: Learning from the Application of Codesign on Library Service Design Projects,” Chapter 11, Quality and the Academic Library Reviewing, Assessing and Enhancing Service Provision, pp. 111–120, 2016.

[Stockstrom, Goduscheit, Lüthje, Jørgensen, 2016] Ch. Stockstrom, R. Goduscheit, Ch. Lüthje, and J. Jørgensen, “Identifying valuable users as informants for innovation processes: Comparing the search efficiency of pyramiding and screening”, Research Policy, vol. 45, issue 2, pp. 507-516, 2016.

[Crystal, Ellington, 2004] A. Crystal and B. Ellington, “Task analysis and human-computer interaction: approaches, techniques, and levels of analysis,” Proceedings of the Tenth Americas Conference on Information Systems, in New York, August 2004.

[Guerrero, Lemaigre, Gonzalez, Vanderdonckt, 2008] J. Guerrero, C. Lemaigre, J.M. Gonzalez, and J. Vanderdonckt, “Model-Driven Approach to Design User Interfaces for Workflow,” IS. Journal of Universal Computer Science, vol. 14, no. 19 (2008), 3160-3173.

[Oliveira, Lepreux, Lepreux, Kolski, Seffah, 2014] K. Oliveira, S. Lepreux, Ch. Kolski, and A. Seffah, “Predictive Usability Evaluation: Aligning HCI and Software Engineering Practices,” IHM’14, 26e conf´erence francophone sur l’Interaction Homme-Machine. Lille, France, pp.177-182, Oct 2014.

[Promann, Zhang, 2013] M. Promann and T. Zhang, "Applying Hierarchical Task Analysis Method to Discovery Layer Evaluation," Libraries Faculty and Staff Scholarship and Research. Paper 107, March 2015.

[Gallagher, Prestwich, 2013] P. Gallagher and Sh. Prestwich, “Cognitive task analysis: analyzing the cognitive of gameplay and game design”, Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), pp 1-10, 2013.

[Crandall, Klein, Hoffman, 2006] B. Crandall, G. Klein and R. Hoffman, Working Minds: A Practitioner's Guide to Cognitive Task Analysis, ISBN: 9780262033510, July 2006.

[McRee, Anderson, Wilson, 2010] J. McRee, J. Anderson and R. Wilson, Effective UI: The Art of Building Great User Experience in Software, ISBN-10: 059615478X, 2010.

[Rubin, 2008] J. Rubin, Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests Book, ISBN: 978-0-470-18548-3, May 2008.

[Marcotte, 2011] E. Marcotte, Responsive Web Design, ISBN 978-0-9844425-7-7, 2011.

[gs.statcounter.com, 2016]

http://gs.statcounter.com/#mobile+tablet-os-CN-monthly-201511-201604-bar

http://gs.statcounter.com/#mobile+tablet-os-US-monthly-201601-201605-bar

http://gs.statcounter.com/#mobile+tablet-os-BG-monthly-201601-201605-bar

http://gs.statcounter.com/#mobile+tablet-os-AU-monthly-201601-201605-bar

[Islam, Bouwman, 2016] M. Islam and H. Bouwman, “Towards user-intuitive web interface sign design and evaluation,” International Journal of Human-Computer Studies. Duluth, MN, USA, vol. 86, issue C, pp.121-137, Feb 2016.

[Nielsen, Landauer, 1993] J. Nielsen and T. Landauer, "A mathematical model of the finding of usability problems," Proceedings of ACM INTERCHI'93 Conference, Amsterdam, The Netherlands, pp. 206-213, April 1993.

[Nagel, 2016] W. Nagel, Multiscreen UX Design, Developing for a Multitude of Devices, pp. 39-73, 2016.

[Marshall, Cook, Mitchell, Summerskill, 2015] R. Marshall, Sh. Cook, V. Mitchell, S. Summerskill, V. Haines, M. Maguine, R. Sims, D. Gyi and K. Case, “Design and evaluation: End users, user datasets and personas,” Applied Ergonomics, vol. 46, Part B, Special Issue: Inclusive Design, pp. 311-317, January 2015.

[Encyclopedia Britannica, 2016] Encyclopedia Britannica Inc., Britannica.com. (May 2016)

[Google material design, 2016] Google material design, https://www.google.com/design/spec/material-design/introduction.html

[Lidwell, Holden, Butler, 2010] W. Lidwell, K. Holden and J. Butler, Universal Principles of Design, Revised and Updated: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach through Design Second Edition, ISBN-10: 1592535879, 2010.

[Telstra identity, 2016] https://www.behance.net/gallery/2900727/Telstra-Rebrand

[Google fonts, 2016] https://www.google.com/fonts#QuickUsePlace:quickUse

[Mozilla Style Guide, 2016] https://www.mozilla.org/en-US/styleguide/websites/sandstone/typefaces/

[Wang, Liu, Soyota, 2014] H. Wang, W. Liu and T. Soyata, "Accessing Big Data in the Cloud Using Mobile Devices," Handbook of Research on Cloud Infrastructures for Big Data Analytics, 2014.

[Liu, 2008] Z. Liu, Paper to Digital: Documents in the Information Age, ISBN-13: 978-1591586203, 2008.

[Liu, 2005] Z. Liu, "Print vs. electronic resources: A study of user perceptions, preferences, and use," San Jose, CA 95192-0029, USA, February 2005.

[echarts.baidu.com] http://echarts.baidu.com/

[Cytowic, Eagleman, 2009] R. Cytowic and D. Eagleman, Wednesday is Indigo Blue: Discovering the Brain of Synesthesia, Cambridge: MIT Press. ISBN 0-262-01279-0, April 2009.

Authors' Information

Margarita Atanasova *– PhD Student in Informatics at Plovdiv University "Paisii Hilendarski" and Web Designer at IT company "Programista;", P.O. 236 Bulgaria Blvd, Plovdiv 4003, Bulgaria; e-mail: margarita.hr.atanasova@gmail.com*

Major Fields of Scientific Research: Adaptive user interfaces, User interface modelling, User experience

Anna Malinova *– Assoc. Prof., PhD, Plovdiv University "Paisii Hilendarski", P.O. Box: 236 Bulgaria Blvd, Plovdiv 4003, Bulgaria; e-mail: malinova@uni-plovdiv.bg*

Major Fields of Scientific Research: Software technologies, Programming, Information technologies in mathematics, Wrapping native legacy codes, E-learning

Hristo Krushkov *– Vice Rector and Assoc. Prof., PhD, Plovdiv University "Paisii Hilendarski", P.O. Box: 236 Bulgaria Blvd, Plovdiv 4003, Bulgaria; e-mail: hdk@uni-plovdiv.bg*

Major Fields of Scientific Research: Computational linguistics, Natural language engineering, Edutainment

MULTILEVEL INTERACTION IN SMART CITY PROJECTS

Ihor Osaulenko

Abstract: *Interaction in the Smart City projects is considered in the paper. The essence of the concept of Smart Cities is revealed. Difficulties which are associated with the implementation of the Smart City projects are showed. The importance of interaction of all stakeholders is underlined. Background of the conception of smart cities is analyzed. The features of smart devices, smart home, and smart urban quarter are briefly discussed. Basic conditions that need to implement the conception of performance are determined. The key stakeholders of the Smart City projects are identified. The basic level of interaction in smart city projects is singled out. It refers to the planning level, the level of the project realization, the level of project teams, and the level of the executors. The specific features of interaction at every level are analyzed. Forming project teams as one of the most important tasks of interaction is defined. The non-forceful interaction theory for solve this problem is used. From the position of the theory, the importance of planning of information flow in the project team is grounded. Using the principle of Minimax for distribution of information flows is suggested. Some other features of distributed project teams are presented. Difficulties of separating project management from operating activities in the Smart City projects are displayed. Mechanisms of interaction in the Smart City projects, using decision support systems, are outlined. Some ways for refining the Smart City projects management are proposed.*

Keywords: *project management, non-forceful interaction, smart city, triple helix*

ACM Classification Keywords: *K.6.1 – Project and People Management, I.2.11 – Distributed Artificial Intelligence*

Introduction

The essence of the Smart City conception is effectiveness of city sources management to increase quality of citizenry life, which is based on using modern information and communication technologies. Moreover, this path of cities transformation corresponds to one of the aims of sustainable development approved by United Nation Organization. But now the conception with all its affinity does not include enough details to complete implementation of its principles.

Obviously, practical realization of Smart City projects involves severe difficulties and both state structures and private corporations can do it by own strength. In this condition, all stakeholders need to interact. Hence, search, analysis and ensuring of the most effective forms of interaction are very actual tasks.

The aim of this article is identification of stakeholders of Smart City projects, the levels of interaction separation and its mechanisms justification.

Smart City projects and mechanisms of interaction

The Smart City conception includes several different directions. Among them we can pick out systems like public water supply, sewerage, heating, electric public utility, street lighting. Other important activities are management of financial flows, integrated security system, transport management. Sometimes monitoring of environment, medical care and social service, municipal free net with access points in public place, automatic sorting of domestic waste, charging station for electric vehicle, and solar batteries mounting, are added to this list.

Evidently, the Smart City conception appeared to be a good reason. The first step was the smart devices creation. Their functions are sufficiently multi-various. For example, we can remember the automatic regulator of heat supply depending on the temperature in the room, lighting switch which has reaction to somebody appearance, washing-machine with built-in-function of powder supply according to fouling factor.

Later such devices had an expanse; many of them had joined to networks. Mobile phones to be used for control of these networks were realized. That cause advancing of the idea of the Smart Home. Owner-occupied dwelling or flat in tenement-house may be described as particular cases. The engineering solutions for both cases have not essential distinctions. But the inhabitants in the multi storey block of flats ought to co-ordinate housekeeping with each other.

The smart urban quarter and urban area became the next steps of this conception. People can unite for solution of public problems. The land improvement may be one of the tasks. In particular, rigging of playground by picture monitors and other smart devices will help parents to control the children and provide safety. Another field of action is connected with discovering of joint interests of people, organization of their meetings, information interchange, rendering of mutual aid in domestic jobs. Often, the particularized Web portal serves for more handy intercourse.

Finally, all smart technologies are included in the Smart City conception [Hatzelhoffer, 2012]. But it's developing needs to satisfy many additional conditions. Above all, we ought to take into account high cost of the projects. This is one of the reasons that, in most cases, the existing decisions are characterized as fragmentary or insular. This way, producing the intelligent control, corporations test and advertise their equipment and technologies. Accordingly, the leadership in the projects belongs to these corporations. For participation in more scaled projects they need precise estimation of recoupment of investment.

Another condition applies to all potential users of intelligent technologies. Untrained person cannot use smart devices and intelligent systems. As a result many advanced functions will be not called for. This circumstance demands of special education programs for all categories of citizens.

The next important issue is the protection of intelligent city networks from illegal access [Greenfield, 2013]. Presence of ample quantity of the smart sensors and devices with access to Internet is the aggravating factor of this problem because often such devices are open to injury from attacks and have no mechanisms of signalizing.

As a whole, for success of the Smart City conception in the specific place the appropriate complex strategy ought to be worked out. More or less extent in the strategy will be presented to all foregoing structure levels. But the basic items are priority aims and accessible resources. Definition of the priorities is very actual task. Investigation of stakeholder's opinions is necessary step for search of optimal decision. Moreover, local authorities ought to stimulate people to share they views on the most significant problems, including through Internet. Monitoring information systems are also important source of data. Analysis and summarizing of gathering information allow defining directions of perspective projects. Obviously, simultaneous realization of all Smart City technologies is impossible. According to reveal preferences scarce resources allocation ought to occur.

But summarizing of citizens opinions is not enough to provide successful projects. The broad participation is very important too. Attraction to team-work is the special question; it calls for effective procedures of interaction. Thereupon we purpose to consider the levels of interaction in detail.

Levels of interaction

- ***Planning***.

The Smart City conception is possible to be considered as one of the forms of innovation development. Among effort in field of innovation the *Triple Helix theory* hold much favor. The essence of this theory is interaction of three main driving forces of innovations: Universities, Industry and Government [Etzkowitz, 2008]. So, planning of the Smart City projects needs in coordination of all contacting parties. Practically all citizens are users of products of the Smart City projects. On the other hand, managers, employees, officials, scientists, etc., are citizens too. Also, the capability for projects generation may be determined as one of the criteria of maturity of information-oriented society [Осауленко, 2014]. Hence, the structures of the civil society will be included inside the Triple Helix. On basis of all available information the plan or program for sustainable city's development in general will be result of interaction on the level of the main institutes.

- ***Projects realization***

At this level, several problems ought to be solved. First of all, potential project participants need to be informed about conception of the project. Stakeholders in the circumstances will analyze the conformity of the project aims to their own interests and evaluate probable profit and risks. Depending on results each takes a decision.

But sometimes, one of the actors is very significant for other partners and the project as a whole. Accordingly, they will make an attempt to persuade not agreeing partner to take part in the project. In general, for definition of participants inclination to consensus, non-forceful (non-power) interaction theory is used [Teslia, 2014]. This theory is based on the use of statistical criteria known as information distance. The information distance is determined by the frequency of reaching an agreement in the history of previous relationships of two partners. This parameter can range from 0 (unity with no questions) to 1 (complete antagonism). Obviously, smaller distance between the actors mean more likely to reach a consensus on further joint activities. During formation the project team, we must avoid both extreme situations. Otherwise we get lack of critical approach or permanent confrontation. As a result, for each project, the organizations that take part in it have to be identified.

Taking into account the information distance, it is relevant not only during the forming of the project team. Planning of information flows in the project team depends on this parameter, too. It is expediently to organize the circulation of information and information influences in compliance with minimax principle. Such approach means minimal interaction of participants' pairs with maximal information distance. This mechanism of interaction allows optimizing team-work at the project.

- ***Distributed project team***

Actually structural units of involved in the project organizations interact at this level. It has some features. On the previous two levels mainly direct contacts take place. At this level interaction is through networking. But specificity of the Smart City, projects stipulate for quick change of work stations, for example, during mounting of equipment and lines of communication. At the same time interaction with inhabitants in the form of feedback occurs.

- **Executors**

Besides this, distributed project teams have some others features. Sometimes, process of production needs of replacement the personnel. In this case, new executors must to grow into a role. This process calls for several time and additional mechanism of interaction which provides for corresponding training and quickly allows taking up the duties. Reasoning from that we also can separate the level of executors.

Interaction in the Smart City project

The Smart City projects are characterized by large quantity of modern engineering solutions. At the same time, application of new principles of city management ought to accompany the existing ones. These conditions make it difficult to interact. The importance of integrated planning and coordination increases. Moreover, as a rule, the project does not end with the start of operation of the relevant smart systems. Additionally, it is required to provide training for users at all levels and to ensure optimal modes of intelligent systems. Thus, it is difficult to determine exactly where the project ends and operating activities start. The experience which is gained in these conditions is very valuable for future project management.

The above considerations provide additional grounds for the use of decision support systems in the Smart City projects management. Using this technology for search of optimal decision can ensure several advantages. Among them, the possibility to estimate more variants and efficiency are often remembered [Ghasemzadeh, 2000]. But realization of the preferences needs in appropriate structure.

The ***interaction in the Smart City project with DSS*** is illustrated by the scheme on Figure 1. We can separate several contours of management and corresponding mechanisms of interaction. The first and simplest case is interaction of members of distributed project team through servers of own organization and messages' control system. This method is appropriate for routine coordination. If the task is more significant, interaction will occur through project server, where all information concerned to the project is checked by the project leader and stored in the project database. For solving optimization problems and searching similar decisions, users can call to the project management models and knowledge base. Sometimes during accomplishment of the project, the necessities of partial team replacement or adding participant proceeding appear. In such cases, the project leader can call to the database of partners. If the project needs in additional resources, the proper message to the City Project Office will be addressed.

Also, the Project Office is connected with the Center of City management and City smart systems. This data exchange allows receiving relevant information about exploitation of the smart systems, discovering imperfection and users requests. It is possible; several users to be in readiness to participate in current projects. Such information is recorded in the projects databases.

The priority aims of the City development are in the scope of the Project Office attention, too. It allows to define the most important projects and to make changes in current projects.

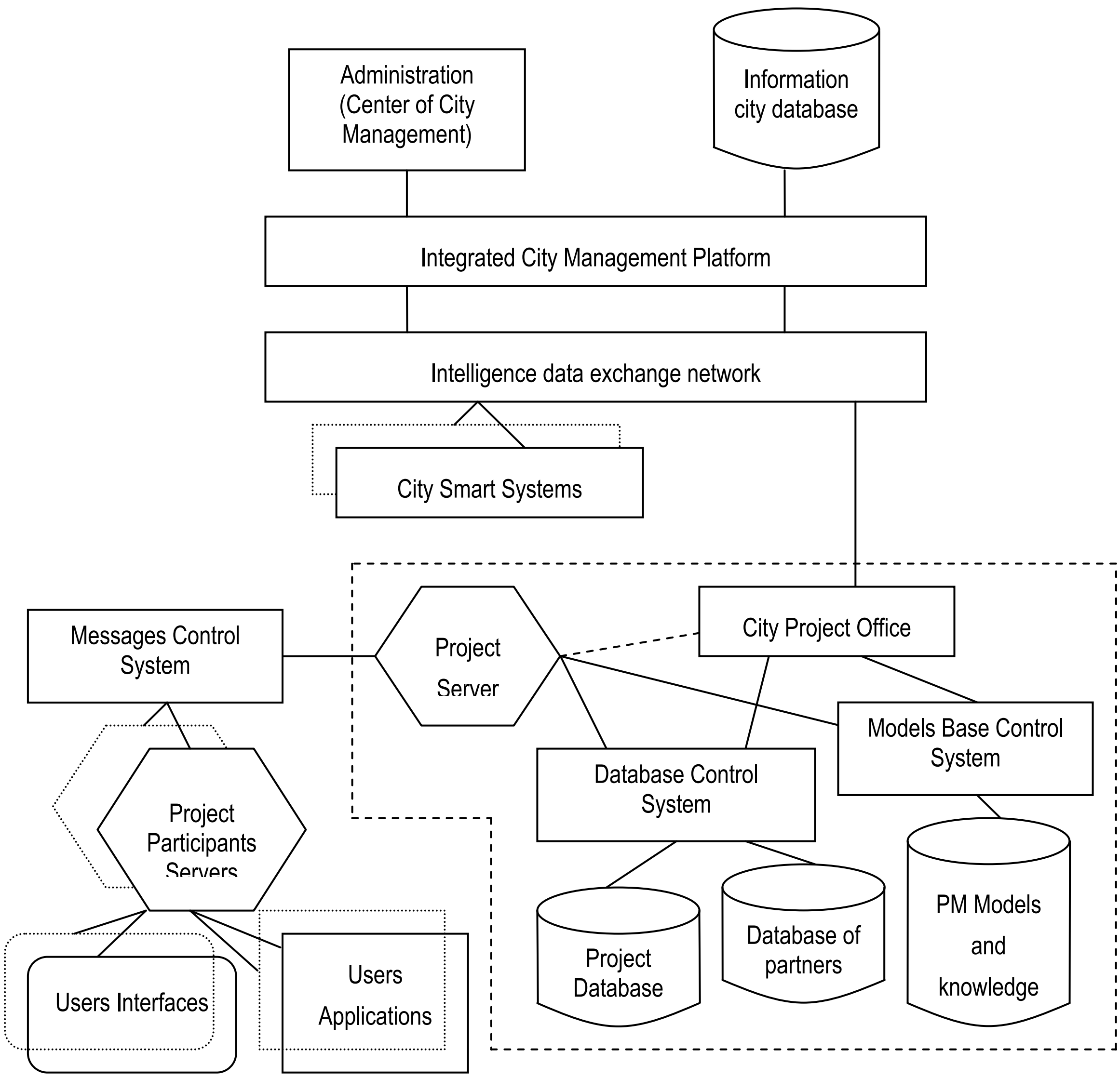


Figure 1. Scheme of interaction in Smart City project using a Decision Support System

Conclusion

Consideration of interaction in the Smart City projects permits to display the problem which during accomplishment of the projects appears. One of the main difficulties is connected to deficiency of detailed elaboration of the Smart City conception. Thereupon City Administration and all other stakeholders together ought to define the foreground tasks for solving.

All citizens are consumers of products of the Smart City projects. In the circumstances, involving of all stakeholders in the Smart City projects planning is very important. It can be obtained by effective mechanism of interaction. All levels of interaction and all stages of the projects are connected to this problem.

As one of the tools of the problem solving, the using of non-forceful interaction theory is proposed. Application of this theory enables the opportunity of efficient selection of the projects participants. It is important to use planning connected with information flows. The Minimax principle in interaction allows obtaining functioning of project team with minimal risk of conflict.

One more significant task of the Smart City projects is developing the professional skills of users. Any smart technology cannot be effective without training the consumers. So, the suitable arrangements are need to stipulate each Smart City project. The protection of the smart systems from illegal access is important task of the projects, too.

Using intelligent technologies in the Smart City project management is very important. Thereupon we have recalled the decision support systems. Possible applications of these systems are rather multifarious. They past experience is accumulated and allow availing project managers of this information. Built-in model is assigned for solving optimization problem.

Advanced approach allows extending traditional functions. One of new application is the module for partners searching and statistics collections. Other possibilities for project management improvement concern the feedback with consumers and more close connection with City Administration. These circumstances can help for success of the Smart City projects.

Bibliography

[Etzkowitz, 2008] H. Etzkowitz. The Triple Helix: University – Industry – Government Innovation in Action. Routledge, 2008.

[Ghasemzadeh, 2000] F. Ghasemzadeh, N. P. Archer. Project portfolio selection through decision support. Decision support systems, Num. 29, pp. 73-88, Amsterdam, 2000.

[Greenfield, 2013] A. Greenfield. Against the smart city (The city is here for you to use). Kindle Edition, 2013.

[Hatzelhoffer, 2012] L. Hatzelhoffer, K. Humboldt, M. Lobecr, C.-C. Wiegandt. Smart City in Practice: Convertive Innovative Ideas into Reality. Jovis, 2012.

[Teslia, 2014] I. N. Teslia.Non-forceful Nature. Monograph. Cherkasy, 2014.

[Осауленко, 2014] И. А. Осауленко. Проектный подход и зрелость информационного общества. International Journal Information Technologies & Knowledge. Vol. 8, Num. 4, pp. 389-391, Sofia, 2014.

Authors' Information

Ihor Osaulenko *– Head of the department of Intellectual Decision Making System, Bohdan Khmelnitsky National University at Cherkassy, Taras Shevchenko avenue, 81, Cherkassy, 18031, Ukraine; email: igrch@ukr.net*

Major Fields of Scientific Research*: Project Management, Information Technologies in Economics, System Analysis*

IN HTTP(S) POTENTIAL TRAFFIC OVERHEAD FOR MOBILE DEVICES

Volodymyr Kazymyr, Andrii Mokrohuz

Abstract: *A mobile device became a powerful computing device, which can provide the same functionality as a personal computer. Increasing amount of different mobile devices leads to increasing of data transferred by application installed on mobile devices. Nowadays, everybody uses a mobile device, due to the fact, that mobile devices are convenient and have big variety of applications available to install. Applications can send data by using the Internet access, which in most cases is provided by Wi-Fi, 3G or 4G networks. This paper considers the most popular application level protocol HTTP (HTTPS) in the Internet. The experiment was carried out to collect HTTP traffic and investigate amount of technical and potentially overhead data that increases overall traffic. The HTTP traffic was grouped by content type. Assumption that headers of HTTP packets contain overhead data was made. Data types with the greatest header to body ratio were identified. High header to body ratio can be non-acceptable for devices with some limitations such as mobile devices. Amount of traffic transferred between a mobile device client and a server plays a significant role in mobile device performance. Therefore, some limitations of mobile devices, in terms of data transferring, were described in the paper as well.*

Keywords: *HTTP, mobile device, packet, header, application level.*

ACM Classification Keywords: *C.2.2. Network Protocols*

Introduction

Mobile devices connected to the Internet are part of our life. A mobile device is a small computing device which has an operation system, runs various applications (software), offers reach functionality, supports interactivity with users, works on battery and has access to the Internet or to any local network. 3G and 4G networks provide anytime access to the Internet. A good example of mobile device is a smartphone or a tablet pc. Any mobile device is a set of hardware and software, which works together.

Mobile device is a small computing device with relatively powerful computing abilities, which makes possible to use it everywhere. The comScore report shows that number of mobile devices' users achieved the number of desktop computers' users in 2014 as shown on Figure 1 [Lella, 2014].

The trend forecasts increasing of mobile devices' user as it is shown on the Figure 1. Mobile devices with the Internet access or network access are considered in the paper. Statistics shows information on the global mobile traffic per month for mobile devices, which can be seen on the Figure 2 [Statista, 2015].

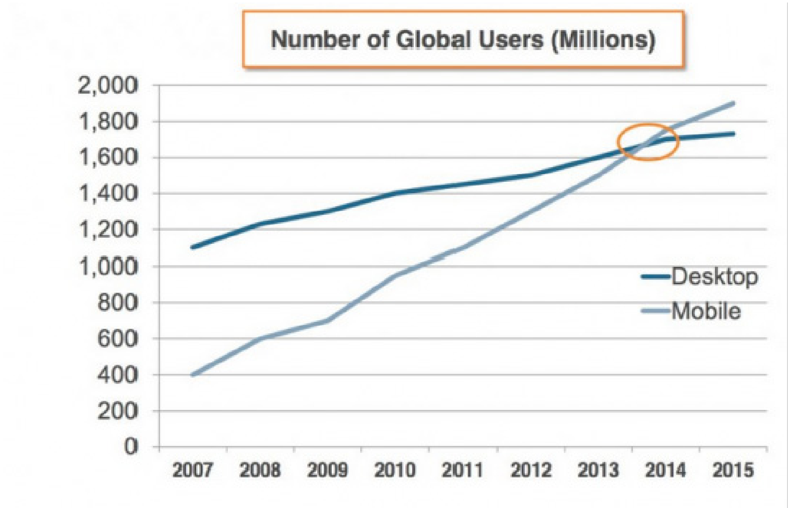


Figure 1 – Number of desktop and mobile devices' users.

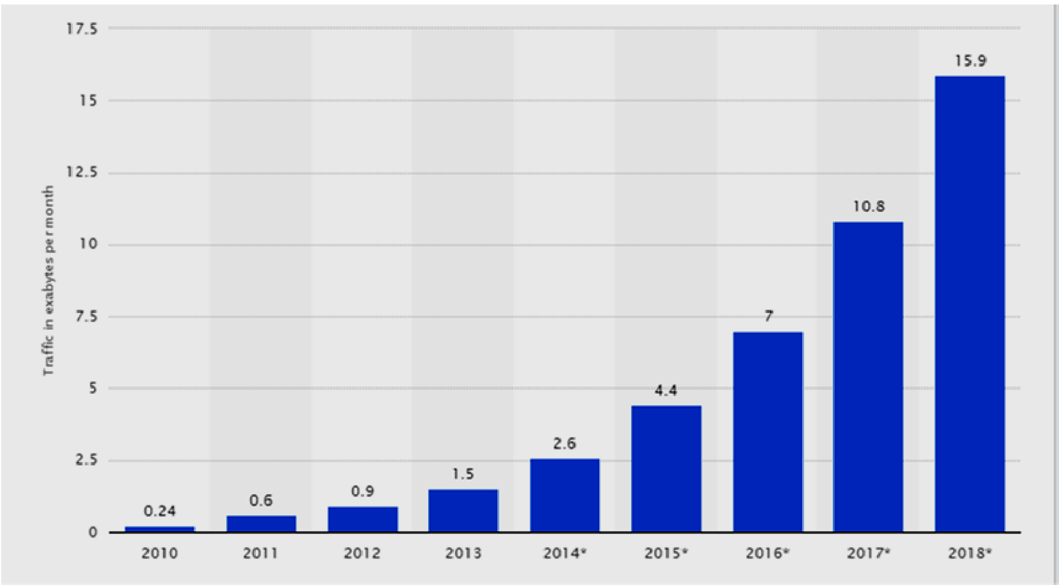


Figure 2 – Global mobile traffic forecast.

In 2015, mobile devices sent and received 4.4 exabytes ($4.4 * 10^{18}$ byte) per month. In 2016, devices will send and receive 7 exabytes ($7 * 10^{18}$ byte) of data per month and this value will grow in the future. Growing amount of mobile devices' users and amount of traffic transferred requires consideration of protocols and mobile devices' limitations (hardware and software) in order to handle increasing amount of data. The paper concentrates on software of mobile devices; especially, it concerns protocols of communication between client and server and touches hardware only in relation with software. We consider these types of data (information) which can be sent and received by mobile devices and be useful to end user:

1) Graphical is the most ancient type of information, which has been implemented for a way to store information about the world in the form of rock paintings, and later in the form of paintings, photographs, diagrams, drawings on paper, canvas, marble and other. The material depicting picture of the real world.
2) Sound (acoustic) type of the information is a kind of musical information. For this type, special method has been devised, which makes storing sound information similarly to graphical information.
3) Text is a method of encoding human speech by using special characters such as letters. Different people have different languages and use different sets of characters to display the text.
4) Numerical type is a quantitative measure of objects and their properties in the surrounding world. It is similar to the text information and special characters (digits) are used to display numbers using the method of encoding. The coding system can be different as well.
5) Video data is a way to save the "live" pictures of the world; it appeared with the invention of cinema.

HTTP for mobile devices

Users of mobile devices and the Internet or network send these types of information from a client to a server and back. Obviously, they send information between pieces of software or applications. Thus, special rules are required to send data from one application to another application. Good example of such rules is a protocol. There are different levels protocols in terms of network communication. These levels are presented by OSI model as shown on Figure 3 [Schwab, 2015].

Data is wrapped into headers and trailers as it seen on Figure 3. This can significantly increase size of a packet transferred via a network. Size of useful data can be smaller than size of technical data on each layer which is required to deliver the packet. The article considers only Application layer protocols. No doubt, one of the most popular protocols is HTTP (HTTPS) in the Internet. Every smartphone or tablet pc has a browser application which uses HTTP to obtain information from a server and send requests on a server.

HTTP protocol allows transferring of these five types of information, which are presented above. The structure of the HTTP protocol is relatively simple, as can be seen on Figure 4. HTTP works in client-server architecture environment: client sends HTTP request and server sends HTTP response back [Fielding, 1999].

Http header defines information, which is required for a client (server) to identify type of a data in a body. Header contains MIME header, which identifies a type of information in HTTP request/response. HTTP protocol can work with all types of information: images, video, text, numbers, sound. Different MIME types exist and they point what information is in a body of HTTP request/response [Fielding, 2014].

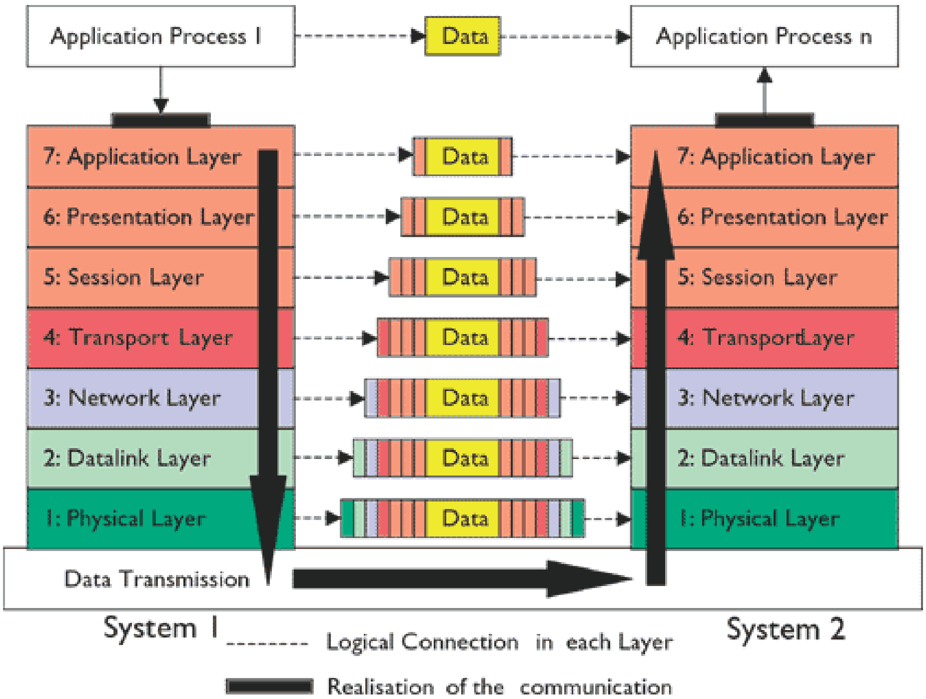


Figure 3 – OSI model and network process communication.

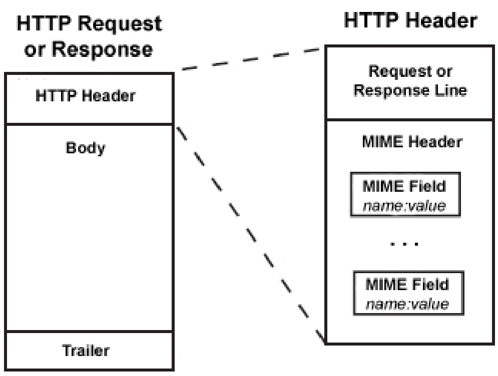


Figure 4 – Structure of HTTP protocol.

Protocol HTTP acts the same way on mobile devices as on normal PCs. However, a mobile device is generally a computing device with some limitations. These limitations make mobile device "special" in terms of transferring data via a network. We consider these limitations for mobile devices.

1. Range and Bandwidth of the Internet connection. 3G or 4G is slower than Wi-Fi or direct cable connection. Many mobile devices cannot have direct cable connection because of its construction. Wi-Fi can be used, but it has limited range. Slower and limited internet connections dictate obvious rule about information transferred from or to mobile device. The

amount of data transferred should be smaller, but this does not mean that loss of data is allowed. Costs of data transferred via 3G or 4G networks are relatively high as well.

2. A mobile device works on a battery power most of the time. 3G and 4G consume battery with bigger intensity. So working time of a mobile device depends on battery life.
3. Usually screens of mobile devices are smaller than devices of normal PC, so developers need to think what information to send to mobile devices' clients more carefully.
4. Security is one of the important issues for communication especially when one of the parties is a mobile device. Security assumes additional overhead of the traffic, so this is another limitation for mobile devices. Because more traffic means less battery life, so functionality of mobile devices is decreased as well.

Client applications, which work on mobile devices, should be implemented in a way, which keeps in mind the mobile devices' limitations. Mobile devices send a lot of traffic as it can be seen on Figure 2. However, this number shows overall traffic including headers and technical requests. In this paper, we have identified amount of technical data in HTTP traffic by conducting the experiment.

Description of the experiment

HTTP packets contains useful for user data and some technical data to make packet deliverable to its destination. Experiment was carried out to identify amount of technical data in different types of data and different content types. Fiddler web debugging tool was used to collect the http traffic on user's computers. Users' PCs were running Windows 7, 8.1 or 10. There were no special rules for experiment. Users lunched the Fiddler application on their computers and were acting as usual during working day (7 hours). Traffic has been provided by users from Ukraine and the United Kingdom. More than 30 000 HTTP packets were captured during the experiment. This experiment was not targeting to capture any specific type of data.

Results of the experiment

All captured data types (image, sound, text and video) are presented in the Table 1.

Table 1. General results of the experiment

Content type (data type) of packets	**Header size bytes sent**	**Body size bytes sent**	**Header size bytes received**	**Body size bytes received**
Without content type technical Data	1763533	751	2094350	331315

application/javascript	434312	166	179746	7795996
application/json	444664	54535	175779	1030410
application /ocsp-response	34774	4822	25791	115011
application/octet-stream	175870	0	101811	10712353
application/pkix-clr	25398	0	21834	1245605
application/x-javascript	548334	0	307252	12850516
Application/zip	406	0	549	0
audio/mpeg	49939	0	31540	661966617
font/woff	35900	0	25138	1978328
image/gif	1868903	55500	926093	41100607
image/jpeg	1753451	0	1283130	67497881
image/png	1101317	0	550675	23231135
Image/svg+xml	30685	0	15960	247044
image/x-icon	54926	0	24232	223253
text/css	347328	0	152731	6843145
text/html	2613740	1006415	1536377	14644210
text/javascript	5492106	1252222	1628782	7476888
text/plain	1657117	190501	696317	59600930
video/mp4	3688	0	2739	23156083
Total	**18436391**	**2564912**	**9780826**	**342047327**

The pie chart with the distribution by amount of the captured packets is presented on Figure 5.

All the data was group according to these data types: text, sound, video and image. This allowed decreasing amount of HTTP data types and presenting the data in a way that is more convenient. HTTP data types were grouped as they are presented in Table 2. Tomasz Bujlow et.al proposed similar traffic classification, however authors concentrated on traffic classes and we split the traffic according to the data types which recognizable by a human. [Bujlow, 2012]

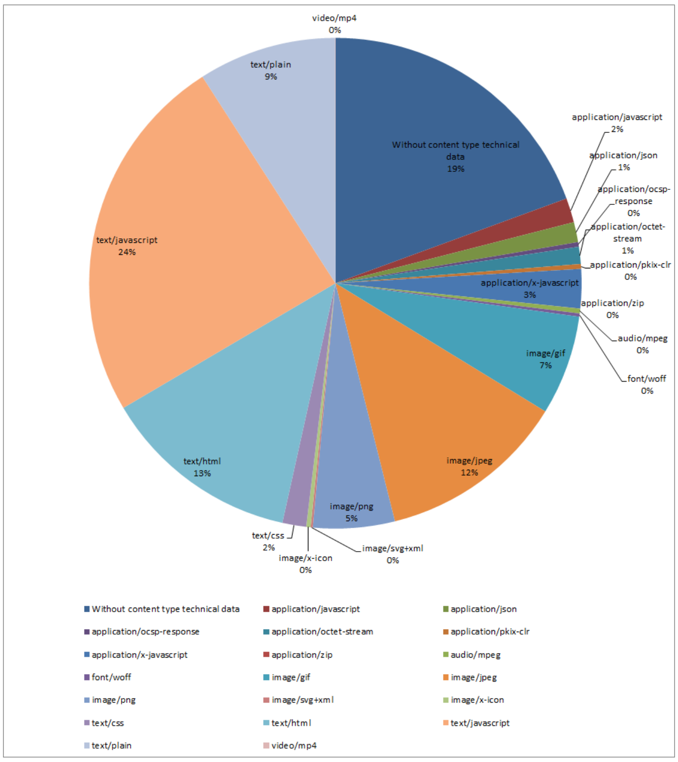


Figure 5 –Distribution of http data by amount of packets.

Table 2. Grouped HTTP data types

General type	**HTTP type**
Text	Without content type technical data, application/javascript, application/json, application/ocsp-response, application/octet-stream, application/pkix-clr, application/x-javascript, text/css, text/html, text/javascript, text/plain, font/woff
Image	image/gif, image/jpeg, image/png, image/svg+xml, image/x-icon
Sound	audio/mpeg
Video	video/mp4

Distribution by amount of packets, header size bytes sent, body size bytes sent, header size bytes received and body size bytes sent is shown on Figure 6. Text and image headers bytes sent and received fill in most of the overall headers traffic. Amount of packets with image and text data takes 99% of overall amount. Despite the sound packets' amount is small, body size received by the client takes 70 % of the overall body size traffic.

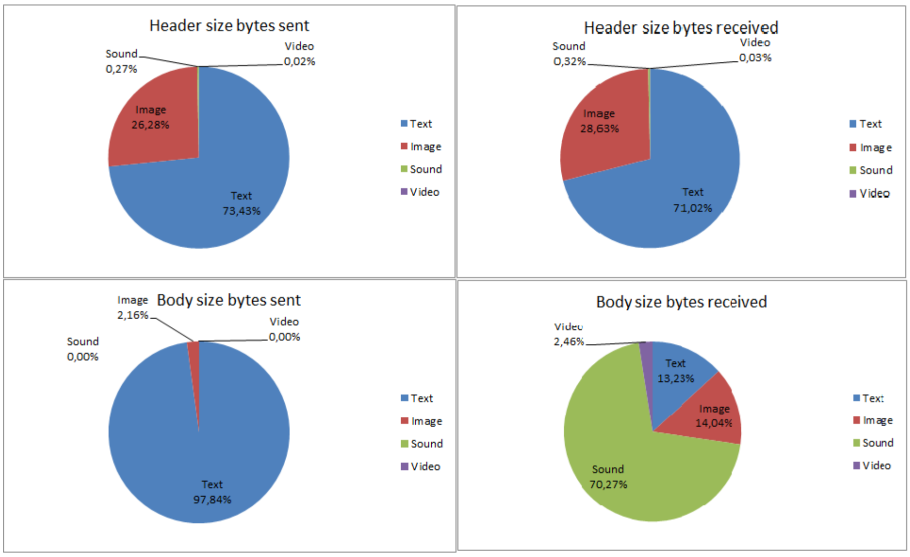


Figure 6 – Graphical results of data distribution by united type of information in bytes.

Distribution of the data types by amount of packets is presented on Figure 7.

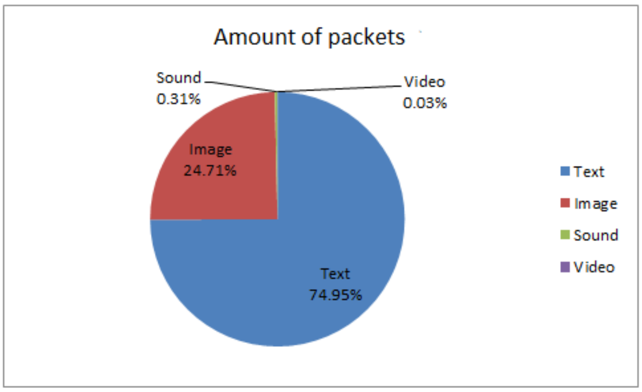


Figure 7 – Data types’ distribution by amount of packets.

Another research, which has been carried out by Pengcheng Jiang et.al, has shown similar distribution by amount of packets. Authors analysed much more network traffic and received results presented on Figure 8. The data was collected from provincial network of well-known ISP (internet service provider) in China [Pengcheng Jiang, 2014].

Type	*Times of occurrence*	*Proportion*
Text	92702334	48.45%
Application	52749385	27.57%
Image	43817095	22.90%
Video	1377169	0.72%
Audio	244219	0.13%
Other	452809	0.24%

Figure 8 – Proportion of different data types.

We have united Text and Application data types into one Text data type and received 74.95% proportion. Pengcheng Jiang et.al received similar results if Text and Application are united 76.02 %. Proportion of image data type is similar as well: 24.71% and 22.90% accordingly [Pengcheng Jiang, 2014].

HTTP headers potentially contain information, which is not necessary to be sent to clients' applications. End users do not need the information, which is inside headers. However, the «useful» information cannot be delivered without headers. HTTP(S) is an application layer protocol, thus software developers usually define what information which client software sends. Sometimes developers use frameworks or libraries provided by third party developers, which create the HTTP requests and responses including headers. Headers for text and image data type take significant part of overall bytes sent and received in comparison to video and sound. The percentage of headers in packets is presented on Figure 9.

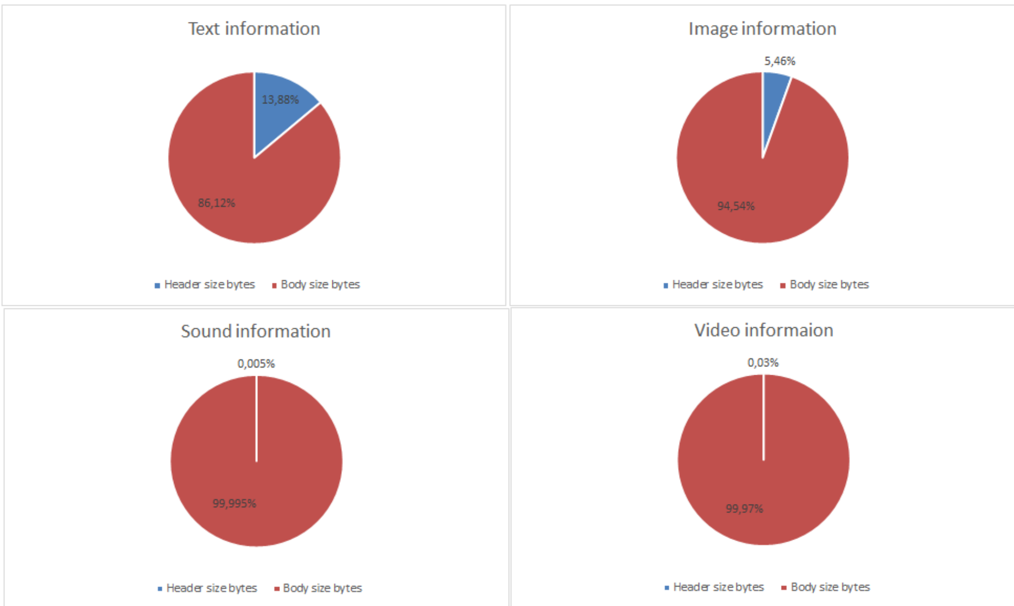


Figure 9 – Ratio of packets' body size to header size by type of information.

The percentage of headers in overall traffic is not as high as in text information as this can be seen on Figure 10.

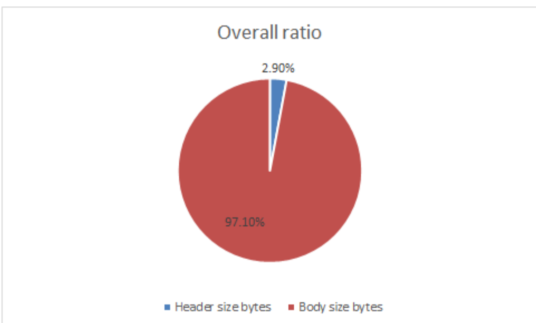


Figure 10 – Body size to header size ration for the data types.

Sound and video packets contains very small amount of headers bytes in comparison to body size. However, text and image headers size in bytes is noticeable. Unfortunately, headers cannot be fully eliminated, but some optimization can be carried out.

Conclusion

Experiment was conducted to identify data types which contain the biggest headers percentage in bytes in HTTP(S) packets. According to the experimental data, Text and Image information contains more technical information than other data types. Mobile devices’ limitations were also considered. Mobile devices’ operational time depends on battery life, mobile networks consumes battery life and decrease time which device can work without charging. The more data client needs to transfer the less time a mobile device can work without charging. Text and image information can decrease the battery life because of potential headers’ overhead.

Results of the experiment allow future development and research in different directions. One of these directions can be consideration of alternative application level protocols. Another one can be improvement technologies and technics directed to decreasing amount of headers’ data. Application level protocols depend on developers, in most of cases a developer decides which data to send. Therefore, some recommendation for developers can be implemented to improve amount of data transferred via HTTP as the most popular application level protocol. These recommendations can be also applied to develop software which does not use HTTP.

Bibliography

[Lella, 2014] Statistics on mobile usage and adoption to inform your mobile marketing strategy. comScore Whitepaper. August 21, 2014.

[Statista, 2015] Global mobile data traffic from 2015 to 2020 (in exabytes per month). Inernet: http://www.statista.com/statistics/271405/global-mobile-data-traffic-forecast.

[Schwab, 2015] Looking behind the automation protocols. Internet: http://www.iebmedia.com/index.php?id=4582& parentid=63&themeid= 255&showdetail=true.

[Fielding, 1999] Hypertext Transfer Protocol -- HTTP/1.1. Internet: http://tools.ietf.org/html/rfc2616.

[Fielding, 2014] Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. Internet: http://tools.ietf.org/html/rfc7231#section-3.1.1.5.

[Bujlow, 2012] T. Bujlow, T. Riaz and J. M. Pedersen, "Classification of HTTP traffic based on C5.0 Machine Learning Algorithm," Computers and Communications (ISCC), 2012 IEEE Symposium on, Cappadocia, 2012, pp. 000882-000887.

[Pengcheng Jiang, 2014] Pengcheng Jiang; Fang Liu; Huan Wang; Chenyu Li, "Characterizing HTTP Traffic of Mobile Internet Services in Provincial Network," in Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2014 Sixth International Conference on , vol.1, no., pp.78-81, 26-27 Aug. 2014.

Authors' Information

Volodymyr Kazymyr *– Dr. Sc., Prof. Chernihiv, National University of Technology, 95, Shevchenko street, Chernihiv-27, Ukraine, 14027; vvkaymyr@gmail.com*

Major Fields of Scientific Research: computer science, information technologies, complicated computer systems.

Andrii Mokrohuz *– Ph.D. Student, Chernihiv National University of Technology, 95, Shevchenko street, Chernihiv-27, Ukraine, 14027; myworkingadr@gmail.com*

Major Fields of Scientific Research: client-server architecture, client-server applications communication, application level protocols.

NEW INNOVATION METHOD FOR SECURE COMMUNICATION BY WIRELESS SENSOR NETWORKS

Irma Aslanishvili

***Abstract**: A sensor is a device that detects events or changes in quantities and provides a corresponding output, generally as an electrical or optical signal. The sensor has to do the following tasks: Give a digital signal, be able to communicate the signal, be able to execute logical functions and instructions. Sensors are used in everyday and they are in everywhere in our life. The objective of “new innovation Method to Secure communication for Wireless Sensor Networks“ is to provide a collection of high-quality research papers in signal processing for Computer sensor systems and Computer Sensor Networks. This innovation Method motivated by the idea of developing the high effective sensory systems for monitoring of environmental pollution.*

***Keywords**: Sensor Networks, Wireless, Modeling and Analysis, MANET protocols.*

Introduction

We study the innovation method for Wireless networks node and our problem of selecting an optimal route in terms of path availability. We propose an approach to improve the efficiency of reactive routing protocols. Ad hoc networks each node acts as a router for other nodes. The traditional link-state and distance-vector algorithms do not scale well in large MANETs.[Namicheishvili et al, 2011]

New innovative method for Wireless sensor networks are more popular. We investigate the following important problems for the wireless ad hoc environment. We address the problem of group access control for secure group communications in ad hoc networks. Wireless Ad Hoc and Sensor Networks envisioned being self-organized, self-healing and autonomous networks, deployed when no fixed infrastructure is either feasible or cost effective. However, the successful commercialization of such networks depends on the implementation of secure network services, for supporting secure applications. Compared to the existing approaches for infrastructure-based networks, we show that in the ad hoc case, the network topology must take into account in the design of a resource-efficient key management schemes. To conserve energy, we incorporate the node location, the “power proximity” between nodes, the path loss characteristics of the medium and the routing topology, in the key management scheme design. While ad hoc networks offer significant advantages in terms of flexibility and cost, they pose great challenges in realizing secure communications via attack-resistant network functions. Oftentimes, ad hoc networks operate untethered in hostile environments in which case, an adversary may eaves drop communications, attempt to inject false messages into the network, impersonate valid network nodes, or compromise nodes causing them to misbehave. Given that ad hoc networks rely on the cooperation principle, attacks on even a few network nodes can have a significant impact in the overall network performance. Problem Statement Method for New innovative methods to secure communication and distribution of Sensor networks.[Aslanishvili 2012]

We study the problem of enabling nodes of an ad hoc network to determine their location even in the presence of malicious adversaries. A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental condition, and by cooperatively pass their data through the network to a main location. The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. On the other hand, we can distinguish also two kinds of nodes: Aggregator and Device or Sensor/Actuator.

Area monitoring is a common application of WSNs. In area monitoring, the WSN deployed over a sensor field where some phenomenon is to monitor. When the sensors detect the event being to by monitored, the event reported to one of the base stations, which then takes appropriate action. Sensor nodes can by imagined as small computers, extremely basic in terms of their interfaces and their components.

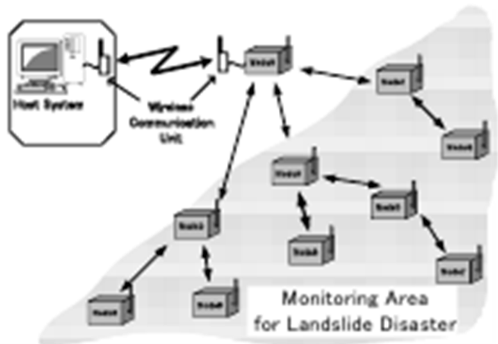


Fig.1 Example of wireless sensory network

This problem will be referred to as Secure Localization. We consider secure localization in the context of the following design goal decentralized implementation, resource efficiency, range-independence, robustness against security threats. Network Model We assume a two-tier network architecture with a set of nodes S of unknown location randomly deployed with a density $q_s$within an area A and a set of specially equipped nodes L we call locators, with known location and orientation, also randomly deployed with a density L << q_s System parametersSince both locators and network nodes are randomly and independently deployed. [Aslanishvili et al 2013] It is essential to select the system parameters, so that locators can communicate with the network nodes. The random deployment of the nodes with a density, is equivalent to a random sampling of the area A with rate P_s. Making use of Spatial Statistics theory, if LHs denotes the set of locators heard by a node s, i.e. being within range R from s, the probability that s hears exactly k locators, given that the locators are randomly and independently deployed, given by the Poisson distribution: If a node receives a beacon transmitted at a specific antenna sector of a locator Li, it has to be included within that sector. Given the locator-to-node communication range R, the coordinates of the transmitting locators and the sector boundary lines provided by the beacons, each node determine its location as the center of gravity (CoG) of the overlapping region of the different sectors.

The base stations are one or more distinguished components of the WSN with much more computational, energy and communication resources. They act as a gateway between sensor nodes and the end user as they typically forward data from the WSN on to a server. The algorithmic approach to modeling, simulating and analyzing WSNs differentiates itself from the protocol approach by the fact that the idealized mathematical models used are more general and easier to analyze.

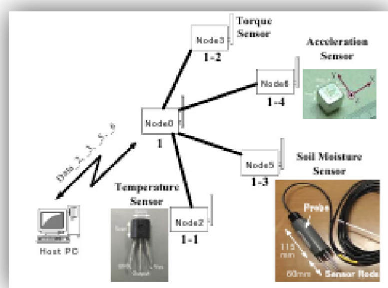


Fig. 2 A simple sensor network for measuring the necessary Parameters.

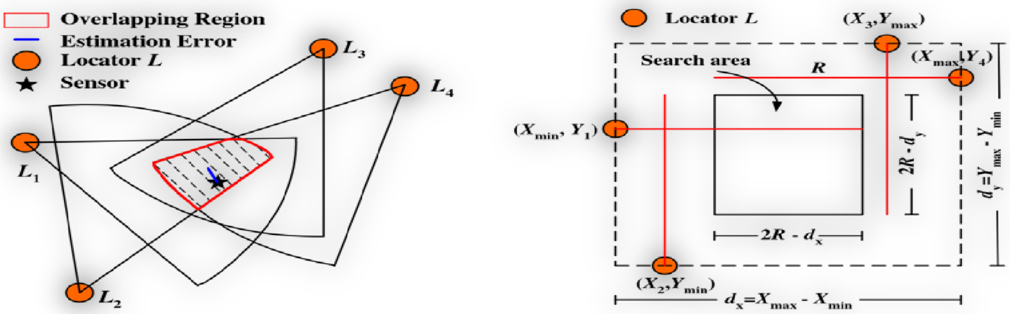


Fig 3. The node hears locators $L_1 \sim L_4$ and estimates its location as the Center of Gravity CoG of the overlapping region of the sectors that include it.

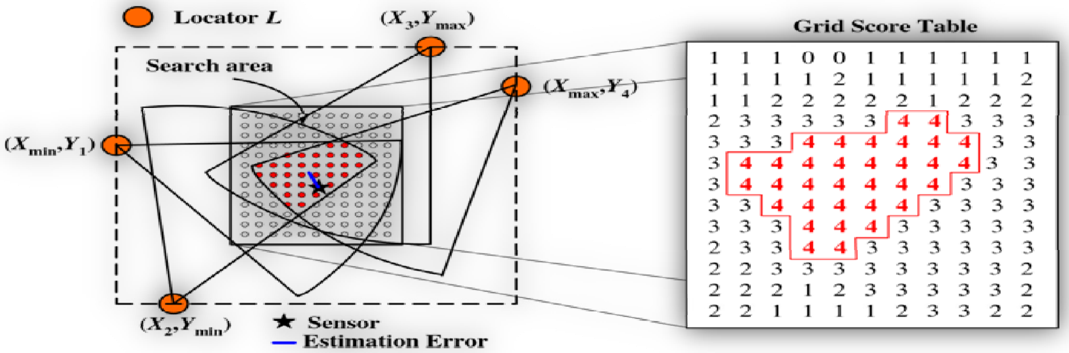


Fig4. The dimensions of the rectangular search area are (2R − d_x)x(2R − d_y) where d_x,d_y are the horizontal distance d_x =X_{max} − X_{min} ≤ 2R.

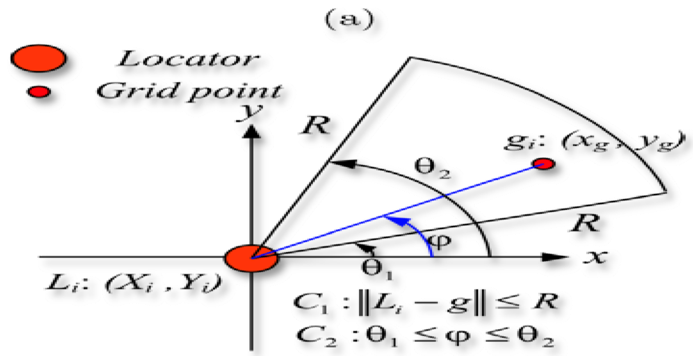


Fig 5. Grid-sector test for a point g

The node estimates its position as the centroid of all grid points with the highest score, [Aslanishvili, 2014] Grid-sector test for a point g of the search area The wormhole attack method - To mount a wormhole attack, an attacker initially establishes a direct link referred as wormhole link between two points in the network. Once the wormhole link is established, the attacker eavesdrops messages at one end of the link, referred as the origin point, tunnels them through the wormhole link and replays them at the other end, referred as the destination The wormhole attack is very difficult to detect, since it is launched without compromising any host, or the integrity and authenticity of the communication.

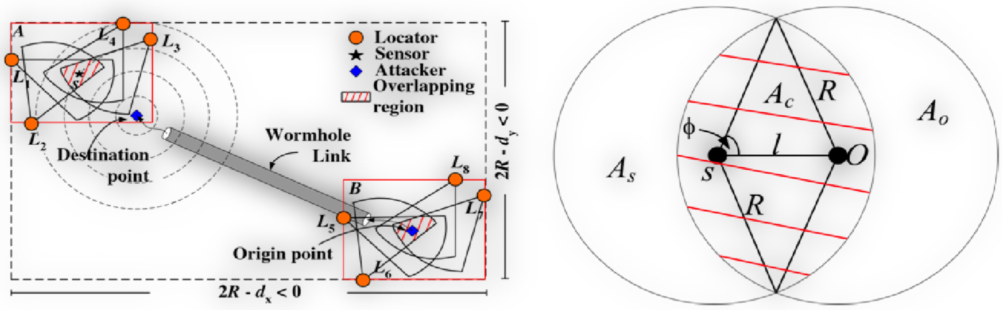


Fig 6. Times its position as the centroid of all grid points with the highest score.

Times its position as the centroid of all grid points with the highest score, Grg of the search are and require each node to frequently recharge its power supply. To overcome the problems associated with the link-state and distance-vector algo-rhythm's a number of routing protocols has been proposed for MANETs. These protocols can be classified into three different groups: proactive, reactive and hybrid. In proactive routing protocols, the routes to all destinations are determined at the start up, and maintained by means of periodic route. [Aslanishvili et al 2015]

Conclusion

New innovation method for Secure communication networks was motivated by the idea of developing the high effective sensory systems for monitoring of environmental pollution, mainly in harsh polluted areas, which can be realized by continuously collecting sensory data from a wireless mobile sensor network deployed in the field. To overcome the problems associated with the link-state and distance-vector algo-rhythm's number of routing protocols have been proposed for MANETs. The relevance of problems particularly pointed out by the environmental dynamism of the shape of fitness function of landscape, which consists of a number of peaks changing width and height in diffuse processes.

Bibliography

[Namicheishvili et al, 2011] Oleg Namicheishvili, Hamlet Meladze, Irma Aslanishvili, Transactions, "Two models for two-hop relay Routing with limited Packet Lifetime", Georgian Technical University, Automated Control Systems, 2011, No1(10), pp. 54-58.

[Aslanishvili, 2012] Irma Aslanishvili, One model for two-hop relay Routing with limited Packet Lifetime the Conference for International Synergy in Energy, Environment, Tourism and contribution of Information Technology in Science, Economy, Society and Education era-7, 2012, ISSN 1791-1133, http://era.teipir.gr

[Aslanishvili ,et al 2013] Energy aware routing model for Wireless and Sensor networks.eRA-8. The contribution of Information Technology to Science, Economy, Society and Education, T.E.I. of Piraeus, Greek; 2013;

[Aslanishvili,et al 2014] Irma Aslanishvili, Three RD Models for Two-Hop Relay Routing With Limited Packets Lifetime In Ad Hoc Networks", International Journal "Information Models and Analyses" Volume 3, Number 3, 2014

[Aslanishvili et al, 2015] Irma Aslanishvili, Paata Kervalishvili, P.Yannakopoulos Novel Methods of Secure Communication and Distribution of Sensor Network https://ener2i.eu/object/event/63/attach/Energy___Innovation_Workshop_2_June_2015_Agenda_ENG_v13a.pdf

Authors' Information

Irma Aslanishvili *– Iv.Javakhishvili Tbilisi State University, Faculty of Exact and Natural Sciences, teacher of Informatics and Computer Sensor Networks; e-mail: Irma.aslanishvili@tsu.ge*

Scientific Research*: General theoretical information research, information systems and computer sensor networks*

TABLE OF CONTENTS OF IJ ITA VOL.23, NO.:1, 2016

TABLE OF CONTENTS OF IJ ITA VOL.23, NO.:2, 2016

TABLE OF CONTENTS OF IJ ITA VOL.23, NO.:3, 2016

TABLE OF CONTENTS OF IJ ITA VOL.23, NO.:4, 2016