WOJSKOWA AKADEMIA TECHNICZNA

BIULETYN

INSTYTUTU SYSTEMÓW INFORMATYCZNYCH

WARSZAWA NR 8/2011

BIULETYN INSTYTUTU SYSTEMÓW INFORMATYCZNYCH

KOLEGIUM REDAKCYJNE
prof. dr hab. inż. Marian Chudy (redaktor naczelny)
prof. dr hab. inż. Andrzej Walczak (z-ca redaktora naczelnego)
prof. dr hab. inż. Andrzej Ameljańczyk
dr hab. inż. Ryszard Antkiewicz
dr hab. inż. Andrzej Najgebauer
dr hab. inż. Tadeusz Nowicki
dr hab. inż. Bolesław Szafrański
dr hab. inż. Kazimierz Worwa
dr inż. Zbigniew Tarapata (sekretarz naukowy)

RECENZENCI
dr hab. inż. R. Antkiewicz
dr hab. inż. A. Chojnacki
prof. dr hab. inż. M. Chudy
dr hab. P. Kosiuczenko
dr hab. inż. T. Nowicki
dr inż. D. Pierzchała
prof. dr hab. inż. Z. Raszewski
dr hab. inż. G. Różański
dr inż. J. Rulka
dr hab. inż. B. Szafrański
dr hab. inż. K. Worwa

ADRES REDAKCJI
Redakcja Biuletynu Instytutu Systemów Informatycznych
Wydział Cybernetyki Wojskowej Akademii Technicznej
00-908 Warszawa, ul. S. Kaliskiego 2
e-mail: biuletynisi@wat.edu.pl
tel.: (22)683-95-04, fax: (22)683-78-58
http://biuletynisi.wcy.wat.edu.pl

Biuletyn ISI jest czasopismem indeksowanym w BazTech
http://baztech.icm.edu.pl/

Tłumaczenie i korekta tekstu w j. angielskim: Wojciech Gilewski
Redakcja techniczna i projekt graficzny okładki: Barbara Fedyna

Wydawca: Instytut Systemów Informatycznych Wydziału Cybernetyki WAT

ISSN 1508-4183

Wersją pierwotną (referencyjną) czasopisma jest wydanie papierowe.

Druk: Remigraf Sp. z o.o., ul. Ratuszowa 11, 03-450 Warszawa

CONTENTS

A Simplified Method for Determining Equivalent Games and Their Solutions

A. AMELJAŃCZYK
aameljanczyk@wat.edu.pl

Institute of Computer and Information Systems
Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw, Poland

The paper proposes a new characteristics of the cooperative game, concerning the possibilities of cooperative players and their coalitions. The function of cooperative game relevance was defined that allows for detailed analysis of the "bargaining power" of individual coalitions. It has been proven that it is a game characteristic function in (0,1) - reduced form that is equivalent to the original game. The introduction of this characteristic allowed a major simplification of computational procedures of determining equivalent games and their solutions.

Keywords: characteristic function of game, cooperative function of relevance, the C-core of the game, Shapley solution, Schmeidler solution, compromise solution.

1. Introduction

Game theory as a branch of applied mathematics dealing with the modeling of conflict situations and their resolution dates back to the forties of previous century.

The first application of game theory focused mainly on military needs, related to the determination of optimal strategies in defense of buildings, methods of escorting supply ships and plane-plane-type chase (escape) strategy [8, 9]. Years of the Cold War strengthened this trend even more. It was in the sixties and seventies that brought the rapid development of models of conflict situations in economic applications [6, 7].

Class of these models is distinguished by a specific character, involving the possibilities of cooperation of the conflicted parties, and thus the need to optimize the strategy of negotiations. This resulted in a need for a completely different approach to modeling the conflict situation. This approach differs significantly from the type of modeling ”advanced” game or in the normal form [10].

Multiplayer cooperative games had a lot of economic applications ranging from models of co-operation of companies, regions and even countries to model the behavior of consumers (clients) on the market (so-called games with a continuum of players [3, 6]).

The uniqueness of the decision-making during conflict situations does not end at their great diversity and specificity of the particular modeling process itself. Even more specifically understood is the essence of the optimization of strategies for dealing with such situations. The classic definition of optimization based on the idea of minimizing or maximizing the real constant function became insufficient. The foundations of the theory of optimization strategies in cooperative and uncooperative games was created by von Neumann and Nash [6, 7, 8]. The idea behind the definition of the concept of optimal decisions in any conflict is understood as a state of equilibrium, naturally ”forcing” participants in the conflict to choose certain behaviors (strategies). In the world literature from this area you can find a lot of modifications of the classical idea of the equilibrium point defined by Nash (won Nobel Prize in 1994) [6, 7, 14]. All proposals, however, preserve ”the idea of stability and unmodifiability” of the decision situation being ”the point of balance”.

After a period of stagnation in the development of game theory in the nineties of last century, we currently see the next stage of revival, this time mainly in the area of economic and ”social” application [6, 7, 14]. A particularly interesting area of applying game theory have become market models in terms of macroscopic and ”micro” scale. An undoubted impetus for this was the next Nobel Prize in game theory, which was given to R. Aumann and T. Schelling in 2005 for their economic use.

2. Mathematical Models of Conflict Situations with the Possibility of Cooperation

Specificity of the situation with the possibility of cooperative decision-making does not consist primarily on determining the strategy of individual pages but the determination of ”areas” and ”conditions” of negotiations leading to the maximization of ’winnings’ (profits) of individual players.

The classic conflict situation model with the possibility of co-operation is called the characteristic function of the game [1, 4, 6].

Formally, the N-person conflict situation with the possibility of cooperation (co-op multiplayer game) is presented in the form:

$$\Gamma = (\mathcal{N}, v)$$

where

$\mathcal{N} = \{1, \ldots, n, \ldots, N\}$ – set of numbers of conflicted sides (set of players),

$v : 2^{\mathcal{N}} \to \mathcal{R}$ – characteristic function of the game. Characteristic function of the game is the mathematical ”aggregate model” describing primarily the ability of the members of the coalition to cooperate, which can be created, that extracts the ”cooperative ability and negotiation”, of each group (coalition) of players.

Definition 2.1 [6, 10]
Characteristic function of a game is called function v, which each possible created coalition $S \subset \mathcal{N}$, assigns the maximum total winnings that their members can attain to distribute among each other, regardless of the proceedings of the other players from the $\mathcal{N} - S$ coalition.
$v(S)$ – total guaranteed winnings for the members of coalition $S \subset \mathcal{N}$
It is assumed that $v(\phi) = 0$.
Symbol $N(\mathcal{N})$ will indicate the set of non-trivial coalitions

$$N(\mathcal{N}) = \{S \subset \mathcal{N} | S \neq \phi, \mathcal{N}, \{n\}, n \in \mathcal{N}\} \quad (2.1)$$

The multiplicity of sets N *and* $N(\mathrm{N})$ are accordingly numbers N *and* $2^N - N - 2$.

<u>Example 2.1</u>
A three-person cooperative game $\Gamma = (\mathcal{N}, v)$
$\mathcal{N} = \{1,2,3\}$ – set of players
$N(\mathcal{N}) = \{\{1,2\}, \{1,3\}, \{2,3\}\}$ – set of non-trivial coalitions.

Characteristic function $v : 2^{\mathcal{N}} \to \mathcal{R}$ is written in the table below

S	Φ	1	2	3	1,2	1,3	2,3	1,2,3
$v(S)$	0	2	4	3	$8\frac{1}{2}$	5	9	12

Determining the value $v(S)$ frequently requires creating a different model for each coalition $S \subset \mathcal{N}$, which in accordance with definition (2.1) will allow determining this value [1, 6, 7, 9, 11]. This stage in modelling is usually complicated and strenuous. By examining the properties of function v we can provide more detail of the conflict situation specificity, ”negotiation power” of each player as well as their ”bargaining power” for possibly establishing a coalition. For example, the fact that $v(\{1,2\}) = 8\frac{1}{2}$ indicates that the mutual action of players with the numbers 1 and 2 bring additional profit

$\Delta v(\{1,2\}) = v(\{1,2\}) - (v(\{1\}) + v(\{2\})) =$
$= 8\frac{1}{2} - (2+4) = 2\frac{1}{2}$

However, $v(\{1,3\}) = 5$ indicates that the trial of the mutual action of players with numbers 1 and 3 adds nothing since $\Delta v(\{1,3\}) = 0$. Creating coalition $S = \{1,2,3\}$ (full agreement) means additional profit to split $\Delta(\{1,2,3\}) = 3$.
Of course the question arises as to how to divide it, so that it all would be ”forced” to cooperate. Further discussion will be devoted to conflict situations, which characteristic functions possess natural qualities of relevance and superadditivity.

Definition 2.2
Game $\Gamma = (\mathcal{N}, v)$ is relevent if $\sum_{n \in \mathcal{N}} v(\{n\}) < v(\mathcal{N})$ occurs.

Definition 2.3
Game Γ is superadditive if

$$v(S) + v(T) \leq v(S \cup T), S, T \subset \mathrm{N}, S \cap T = \phi$$

occurs. Additional properties that often characterize real conflict situations can be, inter alia: convexity, additivity, constancy [10, 12, 13] etc.

3. Multiplayer Cooperative Game Solutions

A multiplayer cooperative game solution consists in determining earnings (payments) to each player from set $\mathcal{N}$ in such a way so that the proposal did not ”affect” any player let alone any of the coalitions to withdraw from the proposal. In addition, it is assumed that the total amount for distribution must be in line with the so-called Pareto condition [10, 12].

Definition 3.1
The set of admissible solutions in game $\Gamma = (\mathcal{N}, v)$ are called a set of imputations $I(v)$

$$I(v) = \{x = (x_1, \ldots, x_n, \ldots, x_N)\} \in \mathcal{R}^N \left| \sum_{n \in \mathcal{N}} x_n = v(\mathcal{N}), \right.$$
$$x_n \geq v(\{n\}),\ n \in \mathcal{N}\}. \tag{3.1}$$

In the literature you can find many suggestions for solutions of the cooperative game $\Gamma = (\mathcal{N}, v)$

”The common, classical proposals” are: solidarity imputation

$$x^S = (x_1^S, \ldots, x_n^S, \ldots, x_N^S) \in I(v)$$

such that

$$x_n^S = v(\{n\}) + \frac{v(\mathcal{N}) - \sum_{n \in \mathcal{N}} v(\{n\})}{N},\ n \in \mathcal{N} \tag{3.2}$$

and proportional imputation

$$x^R = (x_1^R, \ldots, x_n^R, \ldots, x_N^R) \in I(v)$$

such that

$$x_n^R = \alpha\, v(\{n\}),\ n \in \mathcal{N} \tag{3.3}$$

where

$\alpha = \dfrac{v(\mathcal{N})}{\sum_{n \in \mathcal{N}} v(\{n\})} > 1$ – cooperation benefits coefficient assuming that $v(\{n\}) \neq 0,\ n \in \mathcal{N}$.

Proposal x^S expresses the idea of fair sharing of benefits from cooperating, whereas x^R expresses the idea of dividing the winnings proportionally to each players' own capability.
In example (2.1) the following is achieved:

$$x^S = (3,5,4), \quad x^R = (2\tfrac{2}{3}, 5\tfrac{1}{3}, 4) \tag{3.4}$$

You will notice that these solutions, despite some ”natural logic” are not stable, because players numbered 1 and 2, are seeking to maximize their profits by forming the coalition $S = \{1,2\}$ to receive $v(\{1,2\}) = 8\frac{1}{2}$ to split, which is an amount higher than $x_1^S + x_2^S = 8$ and higher than $x_1^R + x_2^R = 8$.
This will cause the necessity of negotiating since player no. 3, being outside of the coalition $S \subset \mathcal{N}$, would only receive $x_3 = 3$.
Approaches of this type generally do not guarantee the stability of the solution. This is mainly due to the fact that the structure of the definition does not include mechanisms resulting from the ”strength and ability to cooperate” of the possible creation of non-trivial coalitions. Similar properties (despite taking to account possibilities of non-trivial coalitions) also have a Shapley solution [10]. The Shapley solution for many years was considered the best proposal for a ”postulated” solution [2, 10] of multiplayer cooperative games. Among the many postulates on the properties of solutions, the stability postulate plays a special role. It ”demands” the belonging of the defined solution to the set of stable solutions provided for the game set of such solutions is not empty. In the general case, the Shapley solution does not meet this postulate. The classic definition of stable solutions was introduced by von Neumann as follows:

Definition 3.2 [6,10]
Let $x, y \in I(v)$. x will be the better solution than y if such a non-empty coalition $S \subset \mathcal{N}$, exists that

1) $x_n > y_n,\ n \in S$
2) $\sum_{n \in S} x_n \leq v(S)$

This fact can be written in the form of belonging to the relation: $(x, y) \in \succ$
Relation $\succ \subset I(v) \times I(v)$ is not a relation of order since it is not transitive nor antisymmetric.

Definition 3.3 [10, 12]
Let $L(v) \subset I(v)$, and
$\operatorname{dom} L(v) = \{x \in I(v)\} \mid$ exists $y \in L(v)$, that $(y, x) \in \succ\}$.
If $I(v) = L(v) \cup \operatorname{dom} L(v)$ and $L(v) \cap \operatorname{dom} L(v) = \varphi$ then $L(v)$ is called a stable set (a von Neumann-Morgenstern solution).

Definition 3.4 [10, 12]
The C-core of the game Γ (the core of the game Γ) is called a set $C(v)$ of all non-dominated imputations.

The necessary and sufficient conditions for set $C(v)$ to not be empty were given by Bondareva and Shapley [10]. Every element $x \in C(v)$ is stable (is a cooperation equilibrium point) in the sense that no coalition cannot be created, which will efficiently reject this solution. Accepting the solution outside the C-core of the game will mean a loss for a coalition and thus the return to the proposal from the C-core.

4. New Conflict Situation Cooperation Characteristics

The idea of solving a cooperative game through C-core is not perfect due to a frequent possibility of two cases:

a) set $C(v)$ can be for some $\Gamma = (\mathcal{N}, v)$ models, an empty set,

b) set $C(v)$ can be for some $\Gamma = (\mathcal{N}, v)$ models ”very numerous”.

In the literature of this field you can find many other solution concepts of multiplayer cooperative games. The quality of these concepts is generally assessed by the degree of realizability of the set, the so-called decision postulates of a ”good solution definition”. They are the following postulates:

- postulate of existence
- postulate of uniqueness
- postulate of stability.

The first postulate is for the existence of a defined solution for a large class of possible problems, the second postulate requires the existence of exactly one solution, and the third – a guarantee that there is no solution better than the defined. In the event of gaming problems, the postulate of stability means the necessity of a defined solution to belong to set of equilibrium points, unless it is not empty.

The concept of the solution in terms of the C-core, while most acceptable, it is far from fully meeting the above postulates. The game classes with a one-element C-core [2, 4, 5, 10, 12] is relatively ”narrow”.

Among the many cooperative multiplayer game solution concepts in the aspect of these postulates particular attention deserves: the Shapley solution [10], the Schmeidler solution (so-called the game N-core) [5, 6, 13], and a compromising solution [2, 4, 5]. The common disadvantage of these solutions are, however, the very complicated computational procedures of their calculation [2, 5, 10, 13].

A certain way of allowing the simplification of the calculation is to replace the $\Gamma = (\mathcal{N}, v)$ game in question with the (0,1) game – standardized. equivalent to the original game. Solving (0,1) games – standardized is generally much simpler, and the existing theorems allow easy calculation of an equivalent game output solution.

Below we will define additional new characteristics based on the characteristic functions of a game, which will allow for a more detailed analysis of the cooperation specificity of the players, the degree of ”stability” of the situation and above all they will help with the calculation procedures to determine appropriate solutions.

Definition 4.2

The cooperation instability function will be called the $\alpha : 2^{\mathcal{N}} \to \mathcal{R}$ function in the form of

$$\alpha(S) = \frac{\sum_{n \in S} v(\{n\})}{v(S)}, \quad S \subset \mathcal{N}$$

For the case of $v(S) = 0$ we will adopt $\alpha(S) = 0$.

In general, the greater this number is the lesser ”bargaining power” the coalition members possess. If this number for a certain coalition is equal to one, than the coalition has no chance to form and to maintain. Members of this coalition are willing to accept ”any proposal” from the set of imputations, offered by any coalition. It is easily noticeable that for games with a superadditive characteristic function the following occurs $0 \le \alpha(S) \le 1, \quad S \subset \mathcal{N}$.

Definition 4.3

The cooperation essentiality function will be called the $\gamma : 2^{\mathcal{N}} \to \mathcal{R}$ function in the form of

$$\gamma(S) = v(S) - \sum_{n \in S} v(\{n\}), \; S \subset \mathcal{N}$$

The greater the number the more the ”ability to negotiate” do the coalition members have, this is understood as the possibility of additional expectations for higher amounts of additional profits from the cooperation. The possibility of the cooperation in the framework of such a coalition can ”significantly” increase the amount of the final payout to its members.

Definition 4.4

The essential cooperation standardized function will be called

$$\delta(S) = \frac{\gamma(S)}{\gamma(\mathcal{N})}, \; S \subset \mathcal{N}$$

The property of the ”standardization” of this function is less obvious. The following theorem settles this.

Theorem 4.1

For essential, superadditive games the following occurs

$$0 \le \delta(S) \le 1, \; S \subset \mathcal{N}$$

Proof:

$$\delta(S) = \frac{\gamma(S)}{\gamma(\mathcal{N})} = \frac{v(S) - \sum_{n \in S} v(\{n\})}{v(\mathcal{N}) - \sum_{n \in \mathcal{N}} v(\{n\})}, \; S \subset \mathcal{N}$$

Since the game is an essential game the following occurs

$$v(\mathcal{N})-\sum_{n\in\mathcal{N}}v(\{n\})>0$$

Due to the superadditivity of function v we get $v(S)\geq\sum_{n\in S}v(\{n\})$ for every $S\subset\mathcal{N}$

This means that $v(S)-\sum_{n\in S}v(\{n\})\geq 0,\ S\subset\mathcal{N}$

i.e. $\delta(S)\geq 0$

From the superadditivity of function v also results $v(S)+v(\mathcal{N}-S)\leq v(\mathcal{N}),\ S\subset\mathcal{N}$, especially since

$$v(S)+\sum_{n\in\mathcal{N}-S}v(\{n\})\leq v(\mathcal{N}),\ S\subset\mathcal{N}$$

Therefore

$$v(S)\leq v(\mathcal{N})-\sum_{n\in\mathcal{N}-S}v(\{n\}),\ S\subset\mathcal{N}$$

Subtracting from both sides $\sum_{n\in S}v(\{n\})$ we get:

$$v(S)-\sum_{n\in\mathcal{N}-S}v(\{n\})\leq v(\mathcal{N})-\left(\sum_{n\in\mathcal{N}-S}v(\{n\})+\sum_{n\in S}v(\{n\})\right)$$

i.e.

$$v(S)-\sum_{n\in S}v(\{n\})\leq v(\mathcal{N})-\sum_{n\in\mathcal{N}}v(\{n\}),\ S\subset\mathcal{N}$$

and this means that $\delta(S)\leq 1,\ S\subset\mathcal{N}$

ultimately $0\leq\delta(S)\leq 1,\ S\subset\mathcal{N}$

■

Functions $\alpha(S),\gamma(S),\delta(S)$ are simple "converseions" of the base characteristic function $v(S)$ and can have their values easily determined based on the table of function $v(S)$.

Example 4.1

For the game from example 2.1. we get

S	1	2	3	1,2	1,3	2,3	1,2,3
$v(S)$	2	4	3	8 ½	5	9	12
$\alpha(S)$	1	1	1	12⁄17	1	7⁄9	¾
$\gamma(S)$	0	0	0	2 ½	0	2	3
$\delta(S)$	0	0	0	5⁄6	0	⅔	1

Even the superfacial analysis of the value of function $\alpha(S)$, $\gamma(S)$ or $\delta(S)$ allows making preliminary assessments as to the negotiation "tendencies" of the members of the potential coalition. Coalitions for which $\alpha(S)=1$ do not have any chance of stable establishment since it contradicts the interests of its members, which is to maximize profits. The higher the value of the coalition essential cooperation function, the higher the chances of an advantageous negotiation.

5. Strategic Equivalency of Games

Definition 5.1

We say that two cooperative games $\Gamma'=(\mathrm{N},v')$ and $\Gamma''=(\mathcal{N},v'')$ are *S*-equivalent, if a positive k number and N real numbers $(\alpha_1,\ldots,\alpha_n,\ldots,\alpha_N)=\alpha$ exist that for every $S\subset\mathcal{N}$ the following occurs

$$v''(S)=k\,v'(S)+\sum_{n\in S}\alpha_n \tag{5.1}$$

S-equivalency games Γ' and Γ'' will be identified by

$$(\Gamma',\Gamma'')\in R \tag{5.2}$$

Relation R is reversible, symmetrical and transitional. Instead of (5.2) we will be also writing $(v',v'')\in R$.

If games Γ' and Γ'' are *S*-equivalent, then they are isomorphic and reversible [1, 10].

S-equivalent games have sets of solutions $I(v')$ and $I(v'')$ with the same "structure" due to the relation of domination [10].

By knowing the solution of $x''\in I(v'')$ of game Γ'' we can determine the appropriate equivalent solution of $x'\in I(v')$ of game Γ'.

It can be easily shown that

$$x''=k\,x'+\alpha\ , \tag{5.3}$$

where the number $k>0$ and $\alpha=(\alpha_1,\ldots,\alpha_n,\ldots,\alpha_N)$ are the solution to the equation 2^N-1 (5.1).

For the primary game we have: $x'=\dfrac{x''-\alpha}{k}$

Subsequently, games in (0,1) will undergo analysis − in short form (standardized games) as a base for calculating solutions of equivalent games. Using simpler games in this class, algorithms determining standardized solutions, calculation of standard solutions (inter alia Shapley solution, Schmeidler solution, and compromising solutions) it will be easier to calculate the appropriate solutions of the original game.

6. Multiplayer Cooperative Games in (0,1) − reduced form

Definicja 6.1

Game $\Gamma=(\mathcal{N},v)$ will be called a game in (0,1) – reduced form, if:

$v(\{n\})=0$ for $n\in\mathcal{N}$

$v(\mathcal{N})=1$

These games will be called normalized games.

The set of acceptable solutions of a reduced game has the form of:

$$I(v)=\left\{(x_1,\dots,x_n,\dots x_N)\in \mathcal{R}^N \,\middle|\, \sum_{n\in N} x_n=1,\ x_n\geq 0,\ n\in \mathcal{N}\right\}$$

The set of all N-player games in (0,1) – in short form will be indicated by $V(N)$

$$V(N)=\left\{v\in \mathcal{R}^{2^N-N-2} \,\middle|\, 0\leq v(S)\leq 1,\ s\in N(\mathcal{N})\right\}$$

Theorem 6.1 [10]

Every essential game is S-equivalent for exactly one normalized game in (0,1) − reduced form.

Theorem 6.2

Let $\Gamma=(\mathcal{N},v)$ N-player essential and superadditive cooperative game.

Standardized cooperative essential function $\delta(S)$ determines the standardized game $\Gamma'=(\mathcal{N},v')$ equivalent to game $\Gamma=(\mathcal{N},v)$, such that $v'=\delta$.

Proof:

Game $\Gamma'=(\mathcal{N},\delta)$ is a game in (0,1) − short form since

$$0\leq\delta(S)\leq 1,\ S\subset\mathcal{N}$$

which results from theorem (4.1).

Game $\Gamma'=(\mathcal{N},v')$ will be a game that is S-equivalent to game $\Gamma=(\mathcal{N},v)$ if such a number $k>0$ and number α_n, $n\in\mathcal{N}$ exists such that for every $S\subset\mathcal{N}$ the following will occur:

$$v'(S)=k\,v(S)+\sum_{n\in S}\alpha_n \qquad (6.1)$$

This is a set of 2^N-N-1 equations.

Definition (6.1) requires that

1) $v'(\{n\})=0,\ n\in\mathcal{N}$
2) $v'(\mathcal{N})=1$

We get a set of N+1 equations additionally.

1) $k\,v(\{n\})+\alpha_n=0,\ n\in\mathcal{N}$
2) $k\,v(\mathcal{N})+\sum_{n\in\mathcal{N}}\alpha_n=1$

Where

$$k=\frac{1-\sum_{n\in\mathcal{N}}\alpha_n}{v(\mathcal{N})},\ \alpha_n=-k\,v(\{n\}),\ n\in\mathcal{N} \qquad (6.2).$$

i.e. $$k=\frac{1+\sum_{n\in\mathcal{N}}kv(\{n\})}{v(\mathcal{N})},$$

after the next conversion we get:

$$k=\frac{1}{v(\mathcal{N})-\sum_{n\in\mathcal{N}}v(\{n\})} \qquad (6.3)$$

since game Γ is an essential game, then

$$\sum_{n\in \mathrm{N}}v(\{n\})<v(\mathcal{N})$$

And this means that $k>0$.

By substituting (6.3) in (6.2) we get:

$$\alpha_n=\frac{v(\{n\})}{\sum_{n\in\mathcal{N}}v(\{n\})-v(\mathcal{N})},\ n\in\mathcal{N} \qquad (6.4)$$

By substituting (6.3) and (6.4) in (6.1) we get:

$$v(S)=\frac{v(S)}{v(\mathcal{N})-\sum_{n\in\mathcal{N}}v(\{n\})}-\frac{\sum_{n\in S}v(\{n\})}{v(\mathcal{N})-\sum_{n\in\mathcal{N}}v(\{n\})}=$$

$$=\frac{1}{v(\mathcal{N})\sum_{n\in\mathcal{N}}v(\{n\})}\left(v(S)-\sum_{n\in S}v(\{n\})\right)=$$

$$=\frac{\gamma(S)}{\gamma(\mathcal{N})}=\delta(S),\ S\subset\mathcal{N}$$

■

Example 6.1

For game $\Gamma=(\mathcal{N},v)$ from example (2.1) calculate the S-equivalent game in (0,1) − in short form.

The table below presents the characteristic game function and cooperative functions:

S	1	2	3	1,2	1,3	2,3	1,2,3
$v(S)$	2	4	3	8½	5	9	12
$\gamma(S)$	0	0	0	2½	0	2	3
$\delta(S)$	0	0	0	⅚	0	⅔	1

Game $\Gamma'=(\mathcal{N},v')=(\mathcal{N},\delta)$ is a game in (0,1) – reduced form, S- equivalent to game $\Gamma=(\mathcal{N},v)$.

By having any solution x' of a normalized game it is very easy to calculate the equilibrium solution x of the "output" game $\Gamma=(\mathcal{N},v)$.

It is easy to show that

$$x'_n=\gamma(\mathcal{N})x''_n+v'(\{n\}),\ n\in\mathcal{N} \qquad (6.5)$$

Example 6.2

For the game from example (2.1), by using the solution of the normalized game equivalency the following solutions can be calculated:

a) solidarity imputation [2, 3]
b) Shapley's vector [6, 10]
c) compromising solution [2, 5]
d) Schmeidler solution (N-core) [5, 6, 13]

By using the theorem 6.2 of the normalized game $\Gamma'=(\mathcal{N},v')$, equivalent to game $\Gamma=(\mathcal{N},v)$ we determine the following:

$$v'(S)=\delta(S),\ S\subset\mathcal{N}$$

S	1	2	3	1,2	1,3	2,3	1,2,3
$\delta(S)$	0	0	0	$5/6$	0	$2/3$	1

For a 3-player normalized game, the procedure of calculating the solutions are very simplified [2, 5, 6]. And so:

a) $\overline{x}^S = (1/N, 1/N, 1/N)$ − solidarity imputation $(N = 3)$

b) $\overline{\Phi}_n(v) = S(v) + 1/2\, v(\mathcal{N} - \{n\}), n \in \mathcal{N}$ − Shapley solution, where

c) $S(v) = \dfrac{2 + w(v)}{6}$ − Shapley's constant $\left(w(v) = \sum_{n \in N} v(\mathcal{N} - \{n\}) \right)$

d) $\overline{x}_n^p = k(v) - v(\mathcal{N} - \{n\}), n \in \mathcal{N}$, where $k(v) = \dfrac{1 + w(v)}{3}$ − compromise constant [2, 6]

e) $\overline{x}_n^N, n \in \mathcal{N}$ − Schmeidler solution [6] is the solution of the sequence problems:
$y \to \min$
at constraints:
$y \geq v(S) - \sum_{n \in S} \overline{x}_n, \; S \in N(\mathcal{N})$
$\overline{x} \in I(v)$

After doing some simple calculations we get the following for the normalized game $\Gamma' = (\mathcal{N}, v')$:

a) $\overline{x}^S = (1/3, 1/3, 1/3)$,
b) $\overline{\Phi}(v) = (3/12, 7/12, 2/12)$,
c) $\overline{x}^p = (1/6, 5/6, 0)$,
d) $\overline{x}^N = (1/6, 5/6, 0)$.

After using the previously specified formulas $x_n = \gamma(\mathcal{N})\overline{x}_n + v(\{n\}), n \in \mathcal{N}$ we get the following solutions of the output game:

a) $x^S = (3,5,4) \notin C(v)$,
b) $\Phi(v) = (2\,3/4, 5\,3/4, 3\,2/4) \in C(v)$,
c) $x^p = (2\,1/2, 6\,1/2, 3) \in C(v)$,
d) $x^N = (2\,1/2, 6\,1/2, 3) \in C(v)$.

From these solutions only the solidarity imputation x^s is not a stable solution. Other proposals lead to stable solutions. Solutions c) and d) correspond to [5].

7. Conclusion

The paper proposes the introduction of new characteristics of co-operative models of conflict situations. They provide a more detailed description of the cooperative and negotiating capacity possible to create a coalition. Such an analysis is essential in determining the proposed settlement of a conflict especially in cases where the C-core of the game is an empty set or if it is a "very large" set [3, 5, 10]. Particularly noteworthy is the definition for the relative cooperation function, which is a "derivative" of the characteristic game function. The function of the relative cooperation $\delta(S)$ is a special characteristic of the possibility of a cooperative coalition of $S \subset \mathcal{N}$. The values of this function are a result of a very simple transformation of the base function game characteristic (definition 4.3) while providing the characteristic function of a normalized game.

It has been shown that this function determines the game equivalency of the original game with the characteristic function in the normalized games class. This allows very simple defining of an equivalent game and determining its solutions.

8. Bibliography

[1] A. Ameljańczyk, *Teoria gier*, WAT, 1978.
[2] A. Ameljańczyk, „Wektorowa optymalizacja w modelach decyzyjnych ze szczególnym uwzględnieniem modeli growych", *Dodatek do Biuletyn WAT*, Nr 2 (1979)
[3] A. Ameljańczyk, „Rozwiązanie wieloosobowej gry kooperacyjnej z C-jądrem mocy continuum", *Biuletyn WAT*, Nr 2, 1977.
[4] A. Ameljańczyk, „Właściwości zbioru superaddytywnych funkcji charakterystycznych gier w (0,1) – zredukowanej formie", *Biuletyn WAT*, Nr 2, 1977.
[5] A. Ameljańczyk, „Rozwiązanie kompromisowe wieloosobowej gry kooperacyjnej względem punktu wyznaczającego z normy Rp (.) a rozwiązanie Schmeidlera", *Biuletyn WAT*, Nr 2 (1978).
[6] R. Aumann, S. Hart, *Handbook of Game Theory with Economic Applications*, Vol. (1, 2, 3) Elsevier North-Holland, 1992, 1994, 2002.
[7] A. K.Dixit, B.J. Nalebuff, *Sztuka strategii,* MT Biznes, 2009.
[8] D. Luce, H. Raiffa, *Gry i decyzje*, PWN, 1964.

[9] M. Malawski, A. Wieczorek, H. Sosnowska, *Teoria gier*, PWN, 2004.
[10] G. Owen, *Teoria gier*, PWN, 1975.
[11] E. Płonka, *Wykłady z teorii gier*, WPŚl., Gliwice, 2001.
[12] A.E. Roth, „Subsolutions and Supercove of Coopratíve Games", *Mathematics of Operations Research*, Vol. 1, No 1, 1976.
[13] D. Schmeidler, *N-core of cooperative games*, MIT, Boston, 2001.
[14] J. Watson, *Strategia – wprowadzenie do teorii gier,* WNT, 2005.

Uproszczona metoda wyznaczania gier równoważnych i ich rozwiązań

A. AMELJAŃCZYK

W pracy zaproponowano nowe charakterystyki wieloosobowej gry kooperacyjnej, dotyczące możliwości negocjacyjnych graczy oraz ich koalicji. Zdefiniowano funkcję korzyści kooperacyjnych gry , która pozwala na szczegółową analizę „siły negocjacyjnej" poszczególnych koalicji. Udowodniono, że jest ona funkcją charakterystyczną gry w (0,1) – zredukowanej formie, równoważnej grze wyjściowej. Wprowadzenie tej charakterystyki umożliwiło znaczne uproszczenie procedur obliczeniowych wyznaczania gier równoważnych i ich rozwiązań.

Słowa kluczowe: funkcja charakterystyczna gry, funkcja korzyści kooperacyjnych, C-jądro gry, rozwiązanie Shapley'a, rozwiązanie Schmeidlera, rozwiązanie kompromisowe.

BIULETYN INSTYTUTU SYSTEMÓW INFORMATYCZNYCH **8** 9−16 (2011)

Properties of the Algorithm for Determining an Initial Medical Diagnosis Based on a Two-Criteria Similarity Model

A. AMELJAŃCZYK
aameljanczyk@wat.edu.pl

Institute of Computer and Information Systems
Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw, Poland

The paper presents analyses of properties of multicriteria decision support mechanisms in the nodes of clinical paths concerning initial diagnosis. The main object of analysis is the two-criteria initial diagnosis model and the developed on its basis a computer algorithm implementation of determining the set of diagnoses from which there is none more probable, and its ranking. A properties analysis of acquired multicriteria diagnoses in terms of distance from the so-called virtual (utopian) diagnosis in the diagnostic area of similarity.

Keywords: disease symptoms, risk factors, initial diagnosis, diagnosis reliability, multicriteria optimization, Pareto set, similarity indicators, similarity relations.

1. Introduction

The collection of works [7, 12, 13], contains detailed results of the tasks relating to decision support in selected nodes of clinical paths conducted within project POIG.01.03.01-00--145/08.

As a result of the analysis made in [7, 12, 31] the concept of building a decision support module was adopted that is based on medical patterns of the patient's health condition and patterns of disease units based on multicriteria analysis. The result of the work that was carried out under the project is the optimization model to generate an initial diagnosis on the basis of observed symptoms of disease and risk factors. This model uses a two-level scheme of assessing the quality of the initial diagnosis. The first level allows the analysis of the severity of the symptoms of disease and risk factors to define the indicators of similarity to the corresponding patterns of diseases occurring in the repository. The second level allows to define two-criteria assessment space similarity of "images" of potential diseases ascertained (during the first visit to the doctor) in the patient's health status [10, 12]. The model was designed so that these functions are replaceable modules and can be defined in different ways depending on the adopted concept of repository modeling (Bayesian networks, rough sets, fuzzy sets, deterministic patterns, spider web patterns etc.) [9, 17, 31] .

The results of the analysis contained in [7, 10, 12] led the author to examine the first approach of using multicriteria decision-making mechanisms in the algorithm determining the diagnostic decisions at the nodes of clinical decision-making paths.

The application element ending the sequence of design work concerning decision support in the nodes of the diagnostic clinical pathways is the multicriteria computer implementation of the algorithm of determining the optimal set of diagnoses in the sense of Pareto and his ranking [3, 11, 12]. This application allows you to perform many interesting tests to investigate the properties of multicriteria mechanisms used in the algorithm and other characteristics of the diagnostic support process. In [11] for the purpose of preliminary testing of the properties of the algorithm the repository was supplied with of expert descriptions for more than twenty diseases. A set of specially generated sets of test results of patients was used. Two opposing diagnostic situations were assumed: the situation characterized by uncertain test results (small degrees of severity of symptoms and risk factors) and the second situation with the ”more expressive” results. Test results will be discussed later in this work.

2. Two-Criteria Diagnostic Model

The decision task of choosing the optimal diagnostic can be defined as follows:

$$\left(\mathcal{M}_o(x), d(x,m), \overline{R} \right) \tag{2.1}$$

where:

$\mathcal{M}_o(x) \subset \mathcal{M}$ – initial estimation of the set of possible diagnostics ($\mathcal{M}$ – repository) for patient $x \in X$ [12].

$d(x,m) = (d_1(x,m), d_2(x,m))$ – two-criteria assessment ”similarity degree” [10, 12] of the patient’s health status $x \in X$ to the disease pattern image $m \in \mathcal{M}_o$

$\overline{R}$ – diagnostic preferences model (called the similarity relation) [3, 5, 7, 12, 13].

The diagnostic preferences models that are most frequently considered are:

- Pareto model[3, 10]
- hierarchical model (1, 2)
- hierarchical model (2, 1)
- pessimist model (optimist) [10]
- collective model (consilium model) [3, 12]

The relation of diagnostic similarities $\overline{R} \subset \mathcal{M} \times \mathcal{M}$ [12] in the general case gives the opportunity to organize a set of potential diagnoses in the context of ”similarity” of the patient's health status model $f(x)$ [13] to the patterns of specific diseases from the set in repository $\mathcal{M}$.

The fact that $(\overline{m}, m) \in \overline{R}$ means that the disease pattern $\overline{m} \in \mathcal{M}$ is more ”similar” to the $f(x)$ model of the patient's health condition $x \in X$ than the disease pattern $m \in \mathcal{M}$. We can interpret it as $\overline{m}$ disease is ”more likely” than disease *m* with currently identified symptoms and risk factors [12, 13].

The optimization model (2.1) means that the similarity of ”the patient’s health condition” with disease $\overline{m}$ is greater than disease *m*, both in the context of observed symptoms and risk factors.

$$\overline{R} = \{(\overline{m}, m) \in \mathcal{M} \times \mathcal{M} \,|\, d(x, \overline{m}) \le d(x, m)\} \quad (2.2)$$

By having the so-called evaluation image of set *Y* of the initial estimate $\mathcal{M}_o(x)$ [12], the problem (2.1) can be written in simple form:

$$(Y, R) \quad (2.3)$$

where:

$$Y = d(\mathcal{M}_o(x)) = \\ = \left\{ (y_1^m, y_2^m) \in \mathcal{R}^2 \,\middle|\, y_1^m = d_1(x,m), y_2^m = \\ = d_2(x,m), m \in \mathcal{M}_o(x) \right\} \quad (2.4)$$

Relation R is a diagnostics similarity model analogically as (2.2).

$$R = \{(y, z) \in \mathcal{R}^2 \, x \, \mathcal{R}^2 \,|\, y \le z\} \quad (2.5)$$

It is a reflexive relation, non-symmetric and transitive – i.e. an order relation [3, 10]. This type of diagnostic preference is called the Pareto model. The solution to problem (2.3) of determining the ”optimal diagnosis” is the so-called domination solution or non-dominated [3, 10]. From the relations properties (2.5) results that, if the domination solution exists than it is unique and it is also a non-dominated solution. Therefore, the solution to problem (2.3), having a practical solution, is a non-dominated solution [3].

The set of non-dominated solutions Y_R^N of problem (Y, R) is defined as follows:

$$Y_N^R = \{y \in Y \,|\, \text{does not exist } z \in Y - \{y\}, \\ \text{that } (z, y) \in R\} \quad (2.6)$$

The opposite image [3.11] of set Y_N^R in the set of initial estimations diagnoses of $\mathcal{M}_o(x)$ is the set of ”non-dominated diagnoses” M_N^R.

$$M_N^R = F^{-1}(Y_N^R) = \left\{ \overset{o}{m} \in \mathcal{M}_o(x) \,\middle|\, d\left(x, \overset{o}{m}\right) \in Y_N^R \right\} \quad (2.7)$$

It is ”a collection of diseases, which none more similar exist in the repository” for the patient's health condition model, with identified symptoms and risk factors.

In other words, in the evaluation space $d(\mathcal{M}_o)$ of image of diseases patterns there are no images ”closer” to the $f(x)$ model of the patient's health condition $x \in X$ than the images of diseases found in set Y_N^R.

The diagnoses ranking proposed in [11, 12] based on the so-called ideal point [3] allows to determine ”the closest diagnosis”, i.e. ”the most probable diagnosis”.

From the general properties of the multicriteria optimization problem results that with the current assumptions in the model, the diagnosis has a property that in repository $\mathcal{M}$ nothing better exists since it is the best.

In the process of diagnosis, as noted earlier, other multicriteria mechanisms can also be used resulting from the adoption of such models as the model of hierarchical preferences, pessimistic (optimistic) or collective.

When using the hierarchical model in the process of diagnosis two cases may occur:

- disease symptoms are more important than risk factors
- risk factors are more important than the disease symptoms

Apart from the opinion of medical experts (the first variant predominates) in the calculation process it does not matter, because according to the properties of the multicriteria optimization problem [3.36], the end result in both cases is included in the M_N^R set (in the Pareto model) and is subject to the ranking in accordance to general principles.

The optimist (pessimist) model as well as the collective (consultation) model was not studied as part of the design work. General properties obtained with these diagnostic results assumptions can be found in [3, 36].

3. Multicriteria Properties of Characteristics of the Diagnostic Process

The main result of the implementation of the algorithm determining the initial medical diagnosis is a set of diagnoses from which there is none more probable as well as its ranking [3, 10]. The credibility of the essential diagnostic procedure and therefore the outcome is a function of many factors, among others, such as ”content” set $\mathcal{M}$ (cardinality patterns of diseases) and individual characteristics of the patient. The patient’s individual properties determine the subjective sensing of the presence and severity of disease symptoms, which in turn affects the so-called ”diagnostic information” patient test results.

From the standpoint of computer support significance in the process of diagnosis information is particularly important on the degree of reliability of results.

The algorithm is designed so that regardless of the ”information quality” of obtained patient test results and regardless of the repository content (a set of disease patterns) leads to the result in the form of a Pareto set and its ranking. However, if there is no disease pattern in set $\mathcal{M}$ for the patient’s disease then determining his/her diagnosis is not possible.

In addition to the diagnosis, information is important about the degree of credibility of the whole procedure as well as ”information importance” obtained results of the patient. This information and many other characteristics can be obtained by analyzing the results of the computer support. Testing the computer application algorithm [11, 13] fully confirmed such a possibility. The adequate ”calibration” of the model and algorithm allow to qualify the results obtained on the basis of, for example, degrees of diagnosis reliability. The subset location of remote images of the initial diagnosis of the estimation of $d(\mathcal{M}_o(x))$ highly depends on the information quality of research results (as far as they are reliable) and on the numbers in set $\mathcal{M}$ containing the repository of disease patterns. The element that characterizes the location of this set is the so-called ideal point $\overset{*}{y}(x)$, which is the greatest lower limit of subset $d(\mathcal{M}_o(x))$ with the adopted model of the similarity relation R (2.3).

Its distance from the virtual pattern image of the disease most probably (that is, from point (0,0)) is a measure of the reliability of the obtained result. Symbol $w(x,\mathcal{M})$ denotes the degree of reliability of the obtained diagnosis on the basis of the results of $f(x)$ [10], the patient $x \in X$ and the number of disease patterns in the repository, represented by set $\mathcal{M}$.

$$w(x,\mathcal{M}) = 1 - q\left((0,0)\overset{*}{y}(x) \right) = 1 - \left\| \overset{*}{y}(x) \right\|_p , \; p \geq 1 \tag{3.1}$$

The role of parameter p in determining the precise form of the standard (3.1) is discussed thoroughly in [3, 36]. In practice, we use p = 2 the most, achieving a simple geometric distance.

Fig.1 shows two diagnostic situations obtained resulting from the algorithm’s actions with the same repository $\mathcal{M}$, and different sets of patient test results (patient $x \in X$ and patient $\bar{x} \in X$ [11, 13]). The diagnostic similarity space, in which the location of images of diseases are studied, obtained resulting from the patients tests is normalized to [0.1] x [0.1]. Depending on the test results (more ”expressive” or less expressive) for each patient receives the appropriate ”focus” points. They create certain characteristic separate subsets.

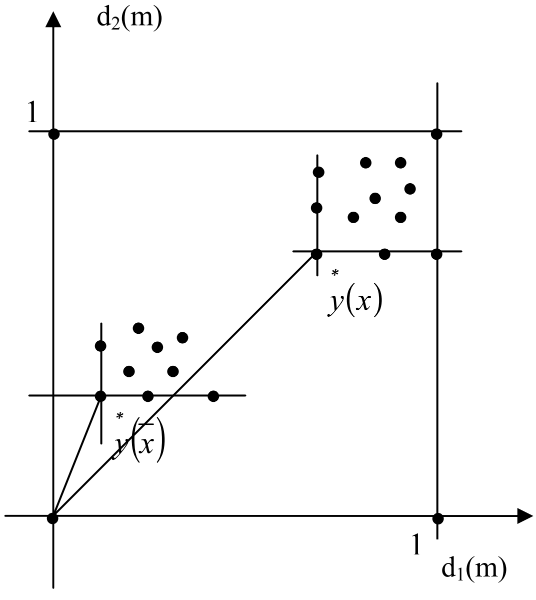

Fig. 1. Diagnostic similarity space

The distance $q\left((0,0),\overset{*}{y}(x)\right)$ in the case of patient $x\in X$ is considerably greater than number $q\left((0,0),\overset{*}{y}(\bar{x})\right)$. These numbers are characterized with reliable diagnoses of particular patients. Therefore we have the following situation:

$$q\left((0,0),\overset{*}{y}(x)\right) >> q\left(\left(0,0,\overset{*}{y}(\bar{x})\right)\right) \to \\ \to w(\bar{x},\mathcal{M}) >> w(x,\mathcal{M}) \tag{3.2}$$

And this means that the diagnosis concerning patient x is considerably less reliable.

However, test of patient $\bar{x}\in X$ have a greater meaning in terms of ”diagnostic information” than test results of patient $x\in X$.

The information degree of test results (in terms of symptoms and risk factors) for specific diseases can be determined by the following figures [10, 12]:

$$i_S(x)=\sum_{s_k^m\in S_O^m(x)} w\left(x,s_k^m\right)\alpha\left(s_k^m\right) \tag{3.3}$$

$$i_R(x)=\sum_{r_l^m\in R_O^m(x)} w\left(x,r_l^m\right)\beta\left(r_l^m\right) \tag{3.4}$$

We can therefore write that the reliability degree of the diagnosis is a function of the content of the repository $\mathcal{M}$ (the exact number of this set) and, information $i_s(x)$ and $i_r(x)$

$$w(x,\mathcal{M})=f\left(i_s(x),i_r(x),|\mathcal{M}|\right)$$

At the same time two of the following remarks are very important:

Remark 1.
The degree of the diagnosis reliability is a non-decreasing cardinality function of repository $|\mathcal{M}|$.

Remark 2.
The degree of the diagnosis reliability is a non-decreasing test results informatibility function in the sense of occurrences of symptoms and risk factors $i_S(x)$ and $i_R(x)$.

There are many other characteristics of these diagnostic inference mechanisms that have important information for the physician. They result mainly from the analysis of the Pareto set properties of obtained diagnostic results. These include:

- ambiguity diagnosis indicator $n(x)$ of patient $x\in X$

$$n(x)=\frac{\max_{y\in Y_N^R}\left(\left|F^{-1}(y)\right|\right)}{\left|F^{-1}\left(Y_N^R\right)\right|} \tag{3.5}$$

- diagnosis clarity $s(x)$

$$s(x)=1-\frac{\left|Y_N^R\right|}{|\mathcal{M}|} \tag{3.6}$$

All these characteristics together with the knowledge of the contents of the set of diagnoses from which there is no more possible and its ranking create a set of information to supporting the process of initial diagnosis.

4. Properties of the Algorithm of Determining the Two-Criteria Initial Diagnosis

The end result of the initial diagnostic process modeling [10] is a model of a decision supported system module and a computer implementation diagnostic algorithm presented in [11]. In this model all the necessary assumptions were included to build an algorithm determining the initial diagnoses based on the establishment done by a physician (during the patient's first visit) symptoms of disease and risk factors along with their degrees of severity. Pattern characteristics of disease entities based on expert data were written to the database of the model [15, 18, 19, 20, 29, 33]. These figures relate to the description of disease entities in the context of associated symptoms and risk factors as well as the degree of their importance in the diagnosis of diseases [10, 12, 13].

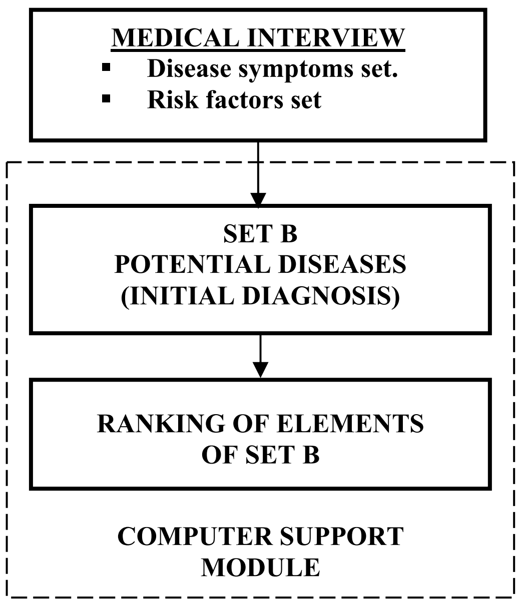

Fig. 2. Computer aided module determining the initial medical diagnosis

Work [11] contains a detailed description of the application architecture, applied information technology and a description of the database model. On the degree of results reliability obtained as a result of the implementation of the algorithm decides, as previously stated, medical history, a patient interview and a set of permanent repository data (expert knowledge). The contents of the set of described patterns of disease entities are of key importance here. Fig.3 shows the schematic processing of medical data

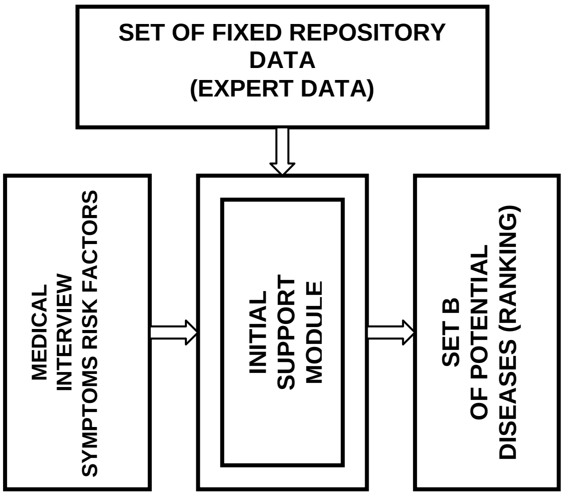

Fig. 3. Scheme of medical data processing

The database is comprised of four tables:

- DESEASE– $\mathcal{M}$ repository (disease list)
- RISK FACTOR OCCURANCE – associative table containing all the possible disease symptoms and risk factors attributable to a given disease, along with an indication of their significance in the diagnosis of disease $m \in \mathcal{M}$
- RISK FACTOR – a table listing the symptoms and risk factors taken into account by the physician during the examination of the patient
- USER – table containing data about the users of the application.

The most important factors determining the quality (effectiveness) of the initial diagnosis has size (cardinality) of set $\mathcal{M}$ (diseases repository). This results from the fact that the developed algorithm compares the image of the patient's condition only with patterns of diseases included in repository $\mathcal{M}$.

The application has been designed so that the extension of tables DESEASE and RISK FACTOR OCCURANCE is easy and the data publicly available [11].

The most important properties of the developed algorithm and application are:

- module replace ability
- ability to visualize obtained results
- ease of creating rankings
- simulation susceptibility
- ability to obtain information about coexisting diseases.

Interchangeability of modules is particularly important if you need to use other similarity functions (distance function) [7] and to compare the results obtained in the context of evaluating the effectiveness of the diagnosis process.

The possibility of visualizing the obtained results can be of great practical importance as an additional diagnostic tool for the family doctor. The graphical representation of a set of potential diagnoses on a computer screen (including a set of diagnoses of which there are none more possible) allows the physician to easily assess the ”significance (reliability) of the diagnosis”. Reliability of the diagnosis (3.1) is a derivative of the distance between patterns of ”diseases that are most probable” than the model of the patient's condition. The smaller this distance is the more reliable the diagnostic process is. Visualization also allows for an initial assessment of the so-called ”clarity of diagnosis” (3.6).

The algorithm allows a very easy way to create a ranking of diagnoses (creating a list of diagnoses from the most likely to the least likely). The position on the list of diagnoses is also linked to a number specifying the distance from the model of the patient's condition.

Comparison of these numbers brings further evidence as to the clarity of the diagnosis and the possibility of the presence of coexisting diseases.

The simulation susceptibility is a very important property of the algorithm, because it facilitates carrying out research on the quality of the diagnosis process. It also gives the possibility of training or testing the diagnostic skills of the physician.

An adequate design of the testing data allows to quickly test the algorithm sensitivity to physician errors during determining the symptoms and risk factors, and mainly their degree of intensity.

The algorithm also enables obtaining information on the occurrence of potential coexisting diseases in the model, even though formally such a possibility was not expected. The possibility of coexisting diseases is mainly due to the cardinality of the Pareto set (the greater the cardinality of this set the likelihood of coexisting diseases is greater). The ”content” of the Pareto set defines the possible ”subsets” of coexisting diseases.

The analysis of the value of the ranking function allows to determine the potential set of diseases that are ”almost as similar” model of the patient's health condition.

5. Conclusion

Presented results in project works [12, 13] and the properties of used diagnostic mechanisms have been confirmed during tests carried out with the application [11]. The use of multicriteria optimization models in initial diagnosis support algorithms allow, depending on the accepted similarity relation model, determining diagnoses taking into account disease symptoms and risk factors. The result of the initial diagnosis is the basis for determining the optimal strategy for additional technical research.

The most important properties of used multicriteria mechanisms to support medical decisions in decision nodes of clinical pathways are:

- the possibility of placing the decision support algorithms in decision nodes of clinical paths
- the guarantee of optimality in terms of Pareto (a set of diagnoses from which there is no other likely in the repository)
- ease in creating a diagnosis ranking of initial diagnoses
- ability to visualize the set of diagnoses of initial disease identification
- replacebility of main algorithm modules (in terms of used patterns, distance function and diagnostic similarity models)
- possibility to extend the basic model with a case of coexisting diseases
- susceptibility simulation algorithm.

Basic research problems to be included in further work are:

- develop a standard of creating (modeling) patterns of disease entities
- model extension with a case of ”coexisting diseases” – the patterns concept of coexisting diseases
- determination algorithm based on the initial medical diagnosis of an optimal strategy for additional technical research
- develop a global three-segment model of determining the final diagnosis (disease symptoms, risk factors, results of specialized tests)
- problem of standardization of medical characteristics obtained from the results of specialized tests.

Basic practical problems to solve are:

- develop a method for computer diagnostic characteristics interpretability of individual disease entities
- develop a method for easy automatic expansion of the list of diseases in the repository
- develop a simple user-friendly interface – the diagnosing medical physician enters diagnostic data (disease symptoms, risk factors, the results of specialized tests and degrees of severity)
- develop a simple ”friendly” visualization module of computer diagnostic decision support results and additional information
- develop a method for calibrating and testing the model and diagnostic algorithms.

6. Bibliography

[1] M. Albin, *Fuzzy sets and their applications to medical diagnosi's*, Berkeley, 1975.

[2] M. Allan, *Crash Course – wywiad i badania przedmiotowe*, Elsevier Urban&Partner, Wrocław, 2005.

[3] A. Ameljańczyk, *Optymalizacja wielo--kryterialna w problemach sterowania i zarządzania*, Ossolineum, 1984.

[4] A. Ameljańczyk, „Analiza specyfiki Komputerowych Systemów Wspomagania Decyzji Medycznych w kontekście modelowania i algorytmizacji procesów decyzyjnych”, I Krajowa Konferencja „Systemy Komputerowe i Tele-informatyczne w Służbie Zdrowia”, Warszawa, 2009.

[5] A. Ameljańczyk, „Matematyczny model przestrzeni życia w komputerowym systemie wspomagania decyzji medycznych”, I Krajowa Konferencja „Systemy Komputerowe i Teleinfor-matyczne w Służbie Zdrowia”, Warszawa 2009.

[6] A. Ameljańczyk, „O pewnej koncepcji modelowania repozytorium medycznego”, POIG.01.03.01-00-145/08/2009, WAT, Warszawa, 2009

[7] A. Ameljańczyk, „Analiza wpływu przyjętej koncepcji modelowania systemu wspomagania decyzji medycznych na sposób generowania ścieżek klinicznych”, *Biuletyn Instytutu Systemów Informatycznych*, Nr 4, 1–8 (2009).

[8] A. Ameljańczyk, „Matematyczne aspekty modelowania pajęczynowego obiektów", *Biuletyn Instytutu Systemów Informatycznych*, Nr 4, 9–16 (2009).
[9] A. Ameljańczyk, „Modelowanie wzorców medycznych w przestrzeni pajęczynowej", VII Konferencja Naukowa „Modelowanie Cybernetyczne Systemów Biologicznych", MCSB2010, Kraków, 2010.
[10] A. Ameljańczyk, „Matematyczny model wspomagania diagnozowania medycznego na podstawie symptomów chorobowych i czynników ryzyka", VII Konferencja Naukowa Modelowanie Cybernetyczne Systemów Biologicznych, MCSB2010, Kraków, 2010.
[11] A. Ameljańczyk, P. Długosz, M. Strawa, „Komputerowa implementacja algorytmu wyznaczania wstępnej diagnozy medycznej", VII Konferencja Naukowa „Modelowanie Cybernetyczne Systemów Biologicznych", MCSB2010, Kraków, 2010.
[12] A Ameljańczyk, „Wielokryterialne mechanizmy wspomagania podejmowania decyzji klinicznych w modelu repozytorium w oparciu o wzorce", POIG.01.03.01-00--145/08/2009, WAT, Warszawa, 2009.
[13] A. Ameljańczyk, „Model formalny informatycznego komponentu wspomagania decyzyjnego ustalenia wstępnej diagnozy medycznej", POIG.01.03.01-00--145/08/2009, WAT, Warszawa, 2010.
[14] ANSI HL7 , http://www.hl7.org.
[15] H.L.C. Beynon i inni, *Interpretacja danych klinicznych w pytaniach i opisach przypadków*, Elsevier Urban&Partner, Wrocław, 2007.
[16] J. Błaszczykowski, K. Krawiec, R. Słowiński, J. Stefanowski, Sz. Wilk, *Wspomaganie decyzji i komunikacji w systemach telemedycznych*, Poznań, 2006.
[17] P. Długosz, „Koncepcja modułu wspomagania decyzji klinicznych w modelu repozytorium z wykorzystaniem metod teorii zbiorów przybliżonych", POIG.01.03.01-00-145/08/2009, WAT, Warszawa, 2009.
[18] R. Douglas Collins, *Algorytmy interpretacji objawów klinicznych*, Medipage, Warszawa, 2010.
[19] F. Kokot, *Diagnostyka różnicowa objawów chorobowych*,WL PZWL, Warszawa, 2007.
[20] F. Kokot, S. Kokot, *Badania laboratoryjne – zakres norm i interpretacja*, WL PZWL, Warszawa, 2002.
[21] J. Makal, „System ekspertowy do wspomagania diagnozy łagodnego przerostu prostaty", *Pomiary Automatyka i Robotyka*, 7–8 (2004).
[22] *Medyczne Systemy Ekspertowe*, http://WWW.computer.privateweb.at/judith/links3.htm
[23] A. Oniśko i inni, *HEPAR I HEPAR II – komputerowe systemy wspomagania diagnozowania chorób wątroby*, XII Konferencja Biocybernetyki i Inżynierii Biomedycznej, Warszawa, 2001.
[24] Z. Pawlak, *Rough Sets, International Journal of Computer and Information Sciences*, vol. 11, 341–356 (1965).
[25] Z. Pawlak, *Systemy informacyjne – podstawy teoretyczne*, WNT, Warszawa, 1983.
[26] Resultmaker, „Workflow patterns of the Online Consultant", version 1.1, Kopenhaga, 2006.
[27] E. Sanchez, „Inverses of fuzzy relations. Application to possibility distributions and medical diagnosis", Proc. IEEE Conf. Decision and Control, USA, 1977.
[28] E. Sanchez, „Medical diagnosis and composite fuzzy relations", *Advances in fuzzy sets theory and applications*, North--Holland, 1979.
[29] W. Siegenthaler, *Rozpoznanie różnicowe w medycynie wewnętrznej*, Tom 1–2, Medipage, Warszawa, 2009.
[30] P. Smets, „Medical diagnosis fuzzy sets and degrees of belief", *Fuzzy sets and Systems*, vol. 5, 1981.
[31] M. Strawa, „Koncepcja modułu wspomagania podejmowania decyzji klinicznych w modelu repozytorium z wykorzystaniem metod sieci bayesowskich", POIG.01.03.01-00--145/08/2009, WAT, Warszawa, 2009.
[32] „*Ścieżki kliniczne jako dynamiczne środowisko dostępu do informacji medycznej pacjenta*", wersja 0.8 Zintegrowany System Informacji Medycznej o Pacjencie, Bielsko-Biała, Kraków, 2008.
[33] The Merck Manual, *Objawy kliniczne*, Elsevier Urban&Partner, Wrocław, 2010.
[34] H. Wechsler, „Applications of fuzzy logic to medical diagnosis", *Proc. Symp. on Multiple – Valned Logic*, Logan 1975.
[35] „Workflow patterns of the Online Consultant", version 1.1, 2006.
[36] P.L. Yu, G. Leitmann, *Compromise solutions, domination structures and Salukwadze's solution*, JOTA, vol.13, 1974.

[37] L.A. Zadeh, „Fuzzy Sets”*, Information and Control*, vol. 8, 338–353 (1965).

Własności algorytmu wyznaczania wstępnej diagnozy medycznej w oparciu o dwukryterialny model podobieństwa

A. AMELJAŃCZYK

W pracy dokonano analizy własności wielokryterialnych mechanizmów wspomagania decyzyjnego w węzłach ścieżek klinicznych dotyczących diagnozowania wstępnego. Głównym obiektem analizy jest dwukryterialny model diagnozowania wstępnego oraz opracowana na jego podstawie komputerowa implementacja algorytmu wyznaczania zbioru diagnoz od których nie ma bardziej prawdopodobnych oraz jego ranking. Dokonano analizy własności uzyskanych diagnoz wielokryterialnych w aspekcie odległości od tzw. diagnozy wirtualnej (utopijnej) w diagnostycznej przestrzeni podobieństwa.

Słowa kluczowe: symptomy chorobowe, czynniki ryzyka, wstępna diagnoza, wiarygodność diagnozy, optymalizacja wielokryterialna, zbiór Pareto, wskaźniki podobieństwa, relacje podobieństwa.

BIULETYN INSTYTUTU SYSTEMÓW INFORMATYCZNYCH **8** 17–24 (2011)

Selected Interoperability Mechanisms Implementation with the Use of the SOA Approach

G. Bliźniuk, R. Dawidczyk
grzegorz.blizniuk@wat.edu.pl

Institute of Computer and Information Systems
Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw, Poland

In this article a short discussion of interoperability was presented according to proprieties of Information Technology (IT) systems. A proposal about some assuring interoperability of IT systems interoperability, so called ”conversion rules” of interoperable data was mentioned. The next, a concept of providing successful and efficient implementation of interoperability mechanisms for cooperating inside IT was described. All main considerations were illustrated in short examples.

Keywords: interoperability condition, conversion rules, SOA.

1. Introduction

Interoperability is a non-functional propriety of cooperating information systems and cooperating information technology systems. The quintessence of interoperability of information systems is the ability to interchange information and make a proper use of it. For information technology systems this condition is understood as interchange and the proper use of data. The reason of this point of view is the assumption about nonphysical integration of systems throughout one new system. Each analyzed system is viewed as a ”black box” with specific interfaces, dedicated for effective cooperation of systems.

Presented understanding of quintessence of interoperability was suggested and shared among many descriptions of this term, from which some are very interesting. According to description from [6], interoperability is understood as an ability of two or more systems or components to exchange and use of information. According to [1], this is described as an ability to efficiently transfer and use unified information across different businesses and information technology systems. This is very important, to look at interoperability throughout the perspective of businesses and business processes. Similar understanding of interoperability can be read in [5].

According to [1] there are certain prerequisites for IT systems to interoperate. One of them is the need for establishing a common domain of data that is supposed to be interchanged (formulas 5−7). That common domain is the collection of data, which shares the same meaning among two independent IT systems. It is a rare situation, when data being the subject of interchange can be exchanged without any additional expenditure on the implementation of a mechanism for their conversion. More likely, as described in article [1], in that kind of situation the use of so-called ”conversion rules” (see formulas 8−10) is necessary. To provide an example of bits of data that have the same meaning but different format, and so requires conversion to be exchanged successfully, one may consider the difference in formats of dates in two given systems. Different date notations differ in numeric or word data representation, separating characters, number of digits or characters representing each period.

In terms of interoperability layers, implementation of a conversion mechanism aims at fulfilling the syntactic and semantic interoperability aspects: ensuring that data that has to be shared (is both correctly sent and received) can be properly interpreted by exchanging participants [2], [1][1].

The most popular approach when implementing an interface for data exchange between independent systems is the SOA concept using the Web-Services standard. Multiple technologies can be used for implementation The protocol that is gaining a particular acceptance is SOAP, which relies on HTTP for data transmission and XML format for structuring messages sent between participants.

[1] See: description of formula 7 in article [1]

Thus, implementation of selected conditions is based on the author's assumption, that interoperable IT systems have interfaces compliant with SOAP and Web-Services principles, and concerns those dimensions only.

Creating software responsible for data conversion for each cooperating system separately is an inefficient, time consuming and costly task. In the process of analysis, all of the necessary conversion rules have to be identified and documented. Afterwards, rules have to be designed, implemented, and tested – for each portion of exchanged data. It is obvious that generated code is not likely to be used as a solution in a different case of cooperation. This situation is illustrated in figure 1, where IT systems use different conversion mechanisms.

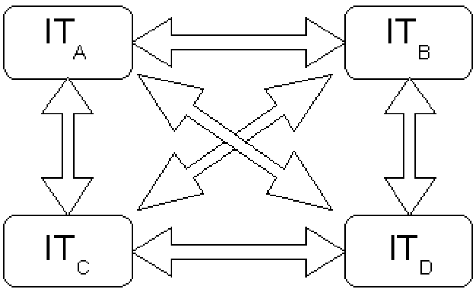

Fig. 1. IT systems exchanging data with a different conversion software for each connection

That analysis led authors to an idea of creating a comprehensive tool, a Converter, which would provide universal data conversion methods. With such software, providing conversion methods for data exchange requires less workload. As a result, creating data conversion routines is less of a challenge. After determining the necessary conversion rules, instead of implementing them, only the configuration of the Converter is required. The use of the Converter (marked as "C") in a situation as shown in figure 1 is illustrated in figure 2.

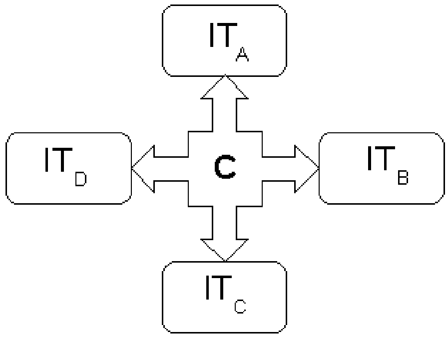

Fig. 2. Converter "C" responsible for data conversion

2. Requirements and Functionalities of the Converter

The main objective of the Converter application is to convert messages from the format provided by the sending system to the format accepted by the receiving system. This is achieved by converting data sent within the SOAP XML document, as well as the XML structures. Figure 3 shows how the Converter is involved in the communication between two IT systems. System "A" is sending a request addressed to system "B". Instead of sending it to the participating system, the request is sent to the application. It should determine the network address of the receiving system and conversions that are to be used based on sender's network address and request name. The request name is the invoked SOAP method. All this information should be provided beforehand as the configuration of the Converter that is stored in database of the application. Afterwards, the converted message is sent to the receiving system, which generates the response document. The response received by the Converter is translated (if necessary) and sent back to the sender i.e. system "A".

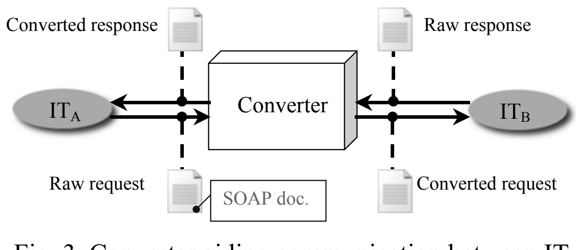

Fig. 3. Converter aiding communication between IT systems

The implemented functionalities of the Converter are shown on use case diagram in figure 4.

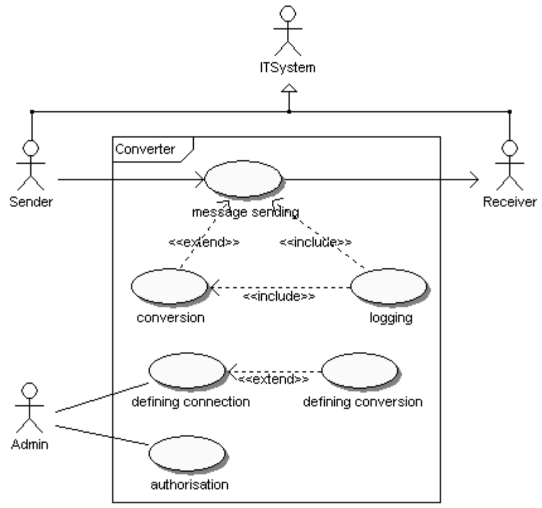

Fig. 4. Use case diagram for the Converter

”Sender” and ”Receiver” are IT systems participating in communication and the most important actors specified on the use case diagram. The use case ”message sending” is associated with those systems. The relation is directional
and indicates, that the sender is the system that initiates the sequence of events, which make up the communication. In SOAP-based architecture, web modules usually provide a number of different methods within a web service. Therefore, for some methods the role of the Converter is simply to forward the received message to the receiver. It is a rare situation, in which data exchanged between systems is compliant and can be sent directly without further workload. For methods that require conversion, the ”message sending” use case is extended by the ”conversion” use case. It represents conversion routines being executed to convert defined data accordingly to the configuration. It is possible to execute more than one conversion routine for a transferred message. In fact, it is desired to use multiple general conversion rules, rather than implementing a single one. It is in this way the general rules implemented in the Converter can be used in different combinations and for different requests. ”Logging” is a use case included in both (already described) use cases. Converter software keeps log information of transferred messages, as well as of errors and exceptional events.

The human actor’s role of ”Administrator” is to provide configuration for the Converter as illustrated by the following use cases:

- definition of the connection between IT systems – network addresses of participants and request name
- definition of conversion – the collection of conversion rules that are supposed to be applied when converting a request or response.

One of the requirements is to provide a convenient way to administrate and configure the application. In order to cope with that need an administration panel has been implemented in the form of a web page. It provides the possibility to add, remove and modify connections between systems, served requests, executed conversion rules and their parameters. Only authorised users have access to the Converter administration panel as shown in the ”authorisation” use case.

The rules implemented within the Converter are divided into two categories; routines converting data wrapped inside XML tags and procedures for converting required structures of the document. Structure modifying rules enable inserting, deleting or modifying names of XML nodes. More complex structural rules allow splitting single nodes with their values into multiple nodes containing portions of original data and vice versa: structure modifications that allow merging multiple nodes into one. On the other hand, data conversion rules allow replacing contents of a given node: modifying date values or formats, doing simple arithmetic or changing/replacing data using regular expressions. For more details and examples refer to [2], chapter 2.2.

3. Implementation Details

The Converter is a tool aimed at supporting communication between systems, which use principles of Web Services and SOAP as a means of data transmission. SOAP in turn relies on web protocols like HTTP, HTTPS or SMTP for data transmission. Hence the author’s decision to implement a translator software as a web application. Among a wide selection of internet technologies, the choice was PHP and Apache HTTP server. The use of such application platform has its advantages. One of those is that there is no need to implement lower level interface for data transmission and negotiation, which allows concentrating the engineering efforts on logical aspects of the application. The environment provides ready solutions for performance and multi-threading aspects. Other advantages are the popularity of the Apache Server and PHP language as well as their open source nature, which is reflected in the amount of documentation and community support, as well as deployment and maintenance costs. Finally, since the fifth revision, PHP is an object-oriented scripting language with decent built-in support of XML documents.

The Converter uses the database to store the configuration. The details of database architecture, including conceptual and physical structure models, are described in [2], chapter 2.5. The postgreSQL was chosen for DBMS. It has the same open source advantages, as PHP and Apache. Additionally, it most likely resembles popular commercial database systems.

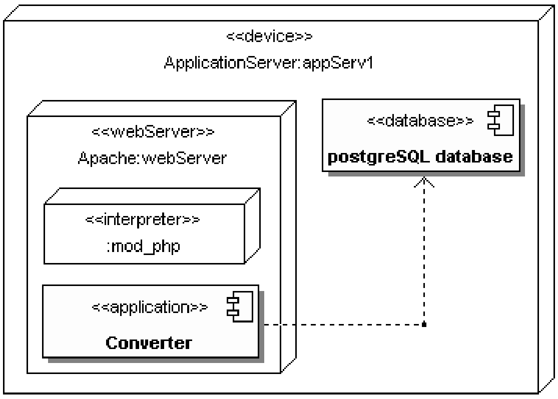

Fig. 5. Deployment diagram for the Converter

The above deployment diagram illustrates the previously described components that make up the structure of an application server. The Converter software was designed to run on a LAPP platform: Linux operating system, Apache HTTP server (<<webServer>>), postgreSQL DBMS (<<database>>) and PHP programming language module ":mod_php" (<<interpreter>>). The <<device>> node is a physical hardware device, on which those components are installed.

The system was modelled and implemented in accordance with object-oriented programming paradigms. UML notation was used for software engineering and development. The most important classes specified, that make up the core of the application, are illustrated in the class diagram, figure 6. The class "Converter" is the fundamental part of the logic. It keeps the information about network addresses of both cooperating systems through the communication process and the name of the request. That information is obtained at the constructor method. Only one instance of this class is created during a single messaging process. Implemented routines coordinate the work of objects used during conversion.

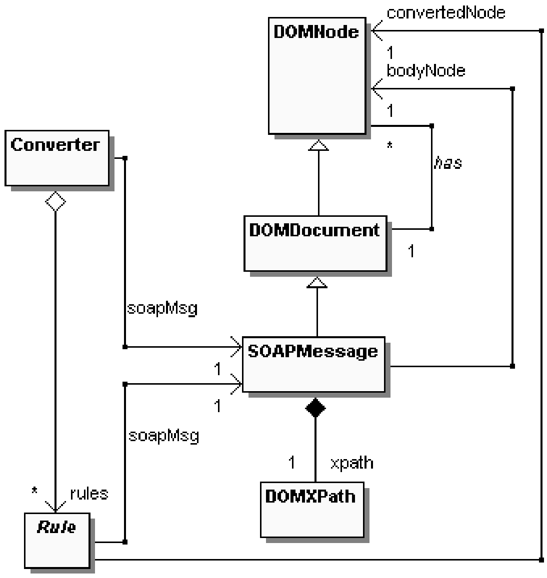

Fig. 6. Core classes of the Converter

The "Converter" class aggregates "Rule" classes. "Rule" is an abstract class. Every conversion rule implemented in the Converter is a type of "Rule". The goal of such a design is to unify the behaviour of all conversion rules implemented in the application. These rules are shown on a diagram in figure 8. Every inheriting class has to provide implementation of an abstract method "run". The body of this method in inheriting entities contains the logic and activities of a given conversion routine.

Both classes are associated with "SOAPMessage" class that is implemented to store the XML message in the form of the DOM object. It provides a convenient way to select and manipulate XML nodes and access their contents or attributes, basing on the built-in PHP classes, such as "DOMNode", "DOMDocument" and "DOMXPath" [7]. The relations result from the built-in DOM API structure.

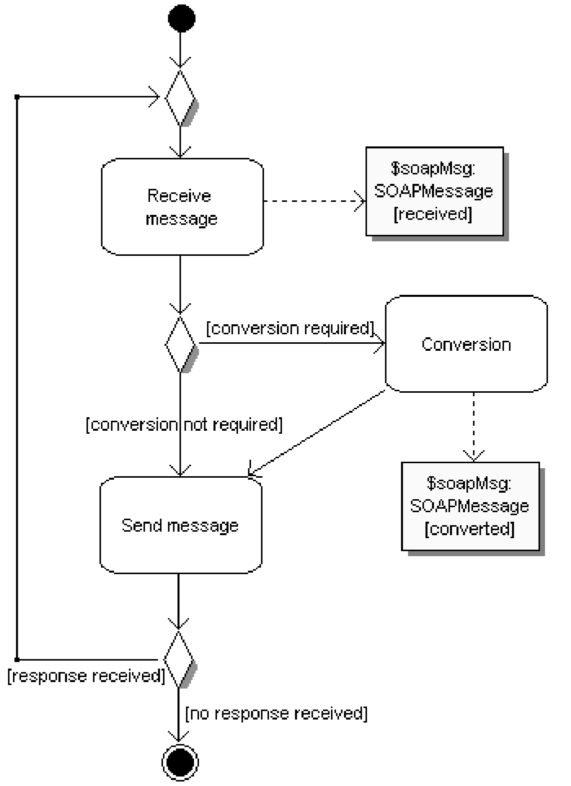

Fig. 7. Messaging and conversion process flow

Activities performed by the application during messaging process are described in the diagram in figure 7. After receiving a request, the XML document is extracted and deserialized from its string form into an object structure. An instance of ”SOAPMessage” is created and, at that moment, it is described as ”received”. The request name is obtained. Afterwards there is a decision, based upon the read configuration data, whether the message contents require conversion. That is a realisation of assumptions introduced in the diagram in figure 4, as a ”conversion” use case that extends the basic message sending. In case a conversion is required, required conversions are executed. Thus, the stored message becomes converted. Regardless of whether the conversion has been made or not, the message is sent to the receiver system. If a response message is received, the messaging and conversion process is invoked again, as shown in the activity diagram.

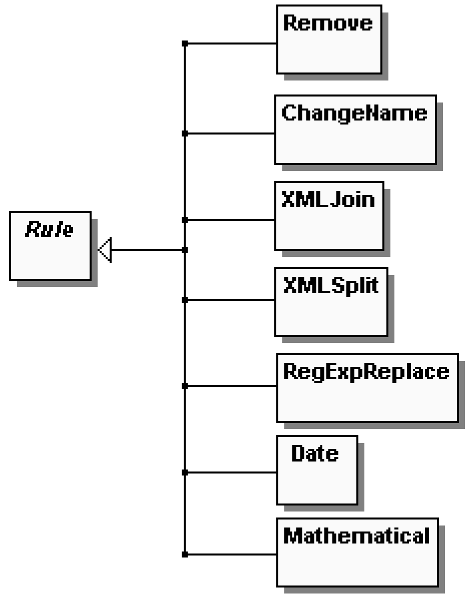

Fig. 8. Kinds of implemented conversion rules

Conversion routines are implemented within classes, the logic of a single conversion rule per single class. When a certain conversion is to be made during the messaging process, the class that contains the rule implementation is constructed and the ”run” method is executed. In order to standardize every conversion class, each one extends the ”Rule” class and provides implementation for abstract methods. Every conversion rule requires the name of the node upon which it is executed, and certain arguments, which control the conversion activity that is to be made. The conversions are divided into two categories: structural and value. Structural conversions are implemented to change the structure of the XML document, whereas value conversions operate on the values of document nodes. Implemented conversion classes are visible in the diagram in figure 8 and described below:

1. ”Mathematical” is a conversion rule that allows simple mathematical formulas to be executed on the contents of a given XML node.
2. ”Date” allows modification of the date information.
3. ”RegExpReplace” – changes the value (contents) of the XML node in accordance with the provided regular expression.
4. ”XMLSplit” – implemented to provide a possibility to split a XML node that contains multiple information into multiple nodes.
5. ”XMLJoin” – opposite to the above: provides a possibility to merge multiple nodes into one.

6. ”ChangeName” – changes the chosen XML node name.
7. ”Remove” – removes the selected XML node.

Examples of some of these rules are presented in the next section, for full reference see [2][2].

Not all of the implemented or used built-in classes are mentioned in this article. Those may be classified as helper classes that aid the application in the messaging process, and include the following:

- ”Database” class – acting as a way to reach the configuration data stored in the database
- ”Logger” class – reporting activities made during the messaging process and encountered exceptions
- ”HttpMessage” class – built-in entity used to send a converted request to the receiving system and to obtain the response content.

For details about the application structure, refer to [2][3].

4. Selected Examples of Messages

To see the effects of conversions done by the Converter, a simple assessment application has been implemented. It sends a declared SOAP message to the Converter, receives the request and generates a response message. The Converter is required to be configured to send the processed request to the assessing application. In effect, it is possible to see messages generated during the messaging process (figure 3), and allows the administrator to test configured conversions in detail before deploying software in a production environment. For more information on testing and the assessment software, refer to [2][4].

The examples of messages presented in this section are not full SOAP messages. They have been trimmed, so that only the contents of the ”Body” section of the SOAP message with the actual data is visible. Furthermore, it is unimportant, whether discussed messages are requests or responses. The aim is to discuss the conversions, based on the comparison of raw and converted messages.

The first example (see Script 1) shows the use of ”XMLSplit” and ”ChangeName” conversion rules. The Converter has been configured to change the name of the node that contains the request/response name from ”getPersonResponse” to ”getPersonNameResponse” as an example of the ”ChangeName” rule. The presented message contains a name and surname of a person. The Converter uses the rule ”XMLSplit” to split that single node into two separate nodes, one for the name and the other for the surname, thus, nodes ”name” and ”surname” are generated. The assumption is that the format of the raw message containing the name data is organized as ”name surname” with a space as a separator.

Raw message:

```
<getPersonResponse>
   <person>Adam Abacki</person>
</getPersonResponse>
```

Converted message:

```
<getPersonNameResponse>
   <name>Adam</name>
   <surname>Abacki</surname>
</getPersonNameResponse>
```

Script 1. Conversion example with ”ChangeName” and ”XMLSplit” rules

The next example illustrates the use of rules ”XMLJoin”, Remove” and ”Date”. The raw message contains information about a date of an exemplary event, and is listed below. Within the message ”eventDate”, event name (node ”name”) and event date (structure ”date”) are sent. The aim is to convert the date structure into a universal format ”CCYY-MM-DD hh:mm:ss” and contain it within a new ”dateTime” node, and to add five hours to the resulting time variable (as an example of the different time zone).

Raw message:

```
<eventDate>
   <name>event</name>
   <date>
      <year>2010</year>
      <month>08</month>
      <day>15</day>
      <time>15:43:00</time>
   </date>
</eventDate>
```

Script 2. Raw message for conversion example with ”XMLJoin”, „Remove” and ”Date” rules.

”XMLJoin” is the first rule invoked for the conversion process. It merges the nodes, which are in the ”date” node structure, and creates a new node ”dateTime” where the result is put.

[2] See chapters: 2.2 and 3.3
[3] See chapter: 2.3
[4] See chapter: 3

The structure of the message after this step is listed below.

Message after ”XMLJoin” rule:

```
<eventDate>
   <name>event</name>
   <date>
      <year>2010</year>
      <month>08</month>
      <day>15</day>
      <time>15:43:00</time>
   </date>
   <dateTime>2010-08-15
15:43:00</dateTime>
</eventDate>
```

Script 3. Message obtained after executing the ”XMLJoin” rule.

This conversion proves, that the order of invoking rules is important. The next steps are to remove the redundant ”date” node (with all the contents) using the ”Remove” rule and, finally, to modify the ”dateTime” node value using the ”Date” rule. The converted message is:

Converted message:

```
<eventDate>
   <name>event</name>
   <dateTime>2010-08-15
20:43:00</dateTime>
</eventDate>
```

Script 4. Fully converted message for the conversion example with ”XMLJoin”, ”Remove” and ”Date” rules.

The last example demonstrates a conversion that is supposed to change the unit in which a speed value is transferred in a message. Conversion rules used are: ”Mathematical” and ”RegExpReplace”. The ”Mathematical” rule is used to calculate the value of speed from mph indication into kph. The ”unit” node is converted with the ”RegExpReplace” rule, which changes the string with the name of the unit from ”mph” into ”km/h”. Messages are:

Raw message:

```
<speed>
   <value>125</value>
   <unit>mph</unit>
</speed>
```

Converted message:

```
<speed>
   <value>200</value>
   <unit>km/h</unit>
</speed>
```

Script 5. Conversion example with ”Mathematical” and ”RegExpReplace” rules.

5. Summary and Further Works

The most important conclusion is the fact that it is possible to implement a tool and mechanisms, which are compliant with definitions described in the introduction of this article, related with [1], [5] and [6]. However, it is important to consider formats and data compliance in cooperating systems through the prism of enabling successful exchange of this data and to both establish and present activities performed throughout the messaging process.

It was possible to construct and implement the Converter application in such a way, that the conversion rules are in fact components that are not inseparable from the rest of the system, but can be added, modified or removed. It is an important advantage, especially when the number of different formats of data that represent the same information is taken into consideration.

Further works, concentrated on increasing the range of functionalities of the Converter may be focused on implementing more conversion rules capable of dealing with different data formats. Those may be general rules, such as described ”XMLSplit” and ”XMLJoin”, or more specific that, for example, extracts the gender from the PESEL[5] number. It is more advantageous to implement less general rules rather than more specific rules, and achieve detailed conversions with a combination of general rules. The desired state of the application is to implement a number of rules that make it possible to convert any format of message into another.

Other improvements can be made within the configuration module of the application [2][6]. Those include: increasing the security of the application, improving the interface by implementing browser side validation. A number of exceptions resulting from configuration errors could be decreased.

Important modification that would improve the configuration module would be the possibility to aid the administrator by uploading and analysing the WSDL document (Web Service description), and generating example requests, responses and examples of converted messages.

[5] PESEL is a governmental unique personal number dedicated for all citizens and residents in Poland

[6] See chapter: 2.6 in [2]

6. Bibliography

[1] G. Bliźniuk, "Thing about Some Assuring Interoperability of Information and Information Technology Systems Conditions"*, Polish Journal of Environmental Studies*, Vol. 18, No. 3B (2009).
[2] R. Dawidczyk, *Implementacja wybranych warunków zapewniających inter-operacyjność systemu informatycznego*, Wydział Cybernetyki WAT, 2011.
[3] G. Booch, J. Rumbaugh, I. Jacobson, Unified Modeling Language User Guide, Addison Wesley, 1998.
[4] Australian Government Information Management Office, *Australian Government Information Interoperability Framework,* April 2006.
[5] Gartner, Preparation for Update European Interoperability Framework 2.0 – FINAL REPORT, 2007.
[6] Institute of Electrical and Electronics Engineers, *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*, New York, NY, 1990.
[7] http://www.php.net/manual/en/intro.dom.php-PHP programming language homepage [visited: 12-04-2011]

Implemetacja wybranych mechanizmów interoperacyności w podejściu SOA

G. Bliźniuk, R. Dawidczyk

W artykule przedstawiono krótką dyskusję warunków interoperacyjności systemów informacyjnych. Zaproponowano i omówiono tzw. regułę konwersji danych dla zapewnienia ich interoperacyjności w systemie informacyjnym. Następnie przedstawiono koncepcję implementacji mechanizmów interoperacyjności w systemie informacyjnym. Rozważania zostały zilustrowane krótkimi przykładami.

Słowa kluczowe: warunki interoperacyjności, reguły konwersji, SOA.

Properties of a Meshless Simulation Method of Diffusion from a Finite Source

T. DAWIDOWICZ
tdawidowicz@wat.edu.pl

Institute of Computer and Information Systems
Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw, Poland

The paper presents an analysis of the properties of a numerical algorithm of a diffusion simulation from a finite source based on a meshless numerical method. The simulation time dependencies and simulation errors have been presented. The range of parameters were indicated for which the method achieves physical results.

Keywords: meshless method, diffusion, simulation.

1. Introduction

The finite element method, which is used very successfully both industrially and academically, has a few limitations. "Quality" interpolation depends largely on the applied mesh. A distorted or very "sparse" grid leads to large errors. In addition, due to the structure of the mesh classical methods, based on it, cannot cope in places of discontinuity [15, 17].

Meshless methods were created for the need to eliminate the problems associated with the basis of approximation on the construction of a fixed numerical mesh.

One of the first meshless methods was proposed by Lucy, Gingold and Monaghan [27, 28] in 1977, a method called Smoth Particle Hydrodynamics Method is primarily used in astrophysics and fluid dynamics, it is based on the construction of the so-called strong form. In 1993 it was successfully applied in solid mechanics [29].

The Galerkin's free element method developed by Belytschko, Y.Y Lu, L. Gu in 1994 was the first based on the global weak form [18]. A year later, a method called Reproducing Kernel Particle Method was developed. Unlike the Galerkin's free element method, which uses moving least squares (MLS) approximation, this method uses wavelets to approximate.

At the end of the 90s the Meshless Local Petrov-Galerkin Method was elaborated.

The main difference between the methods based on the generation of global weak forms, and the Petrov-Galerkin method consists on the generation of weak forms in local domains, and then a full integration within them [1, 2, 7].

Meshless methods originally used in continuum mechanics, very quickly found application in other domains of science. The first publications on the methods for calculating meshless electromagnetic fields appeared in 1998 [13, 20]. Meshless methods were used to solve electrostatic and magnetostatic problems [3, 4, 5, 9, 14, 22,]. In publications [8, 23] a solution method of the Helmholtz wave equation was presented with the help of the meshless method, based on the fundamental solution, and in publication [15] with the help of Galerkin's free elements.

Wavelet analysis also found its use in meshless methods, in publications [10, 11, 12] wavelet analysis in meshless algorithms implementations were presented.

Scope, in which meshless methods found little use is associated with the modelling of dynamic processes.

Most of the currently known solutions refers stationary or quasi-stationary electric or magnetic fields.

The following article presents an analysis of the possibility of applying the meshless algorithm to simulate the diffusion phenomenon.

The basic features, to which attention should be paid to during building the numerical algorithm were identified. A method of analyzing properties and parameters of diffusion simulation based on the meshless algorithm was also proposed.

2. Basic Approximation

Let us consider an unknown scalar function $u(x)$ in domain Ω describing the state of the medium (temperature, density, etc.) .

The basic approximation of the scalar function $u(x)$ in domain Ω can be expressed as [17, 21]:

$$u^h(x) = \sum_{I \in S} \Phi_I(x) u_I(x) \qquad (2.1)$$

where:

$\Phi_I : \Omega \to R$ shape function,

u_I function value in node x_I,

S set of nodes I for which $\Phi_I(x) \neq 0$.

The function shape (2.1) is an approximation, and not an interpolation, because the following occurs:

$$u_I \neq u^h(x_I) \qquad (2.2)$$

Approximation using the method of moving the smallest squares is the basic approximation used in meshless methods [1, 2, 8, 16, 17, 18, 19].

Later in the section the scheme of constructing the shape function for approximation using the method of moving least square (MLS) will be presented.

We are in search for an approximation of an unknown function $u(x)$, in x points called nodes. Let us denote with $u^h(x)$ the approximated function value in point x [8]:

$$u^h(x) = \sum_{j}^{m} p_j(x) a_j(x) \equiv p^T(x) a(x) \qquad (2.3)$$

where: m is the number of base polynomials, $a(x)$ is the vector of coefficients dependent on x.

$$a^T(x) = \{a_0(x) \quad \dots \quad a_m(x)\} \qquad (2.4)$$

In case of one dimension the of base polynomials matrix takes the form of:

$$p^T(x) = \{p_0(x) \quad \dots \quad p_m(x)\} = \{1, x, x^2, \dots, x^m\} \qquad (2.5)$$

Having n function values of $u_1, u_2, u_3, u_4 \dots u_n$ in n nodes $x_1, x_2, x_3, x_4, \dots x_n$ using the equation (2.3) we can express the approximated value of the function in nodes:

$$u^h(x, x_I) = p^T(x_I) a(x) \qquad I = 1, 2, \dots, n \qquad (2.6)$$

Let us build the J functional weighted residual:

$$J = \sum_{I}^{n} w(x - x_I)[u^h(x, x_I) - u(x_I)]^2 = \\ = \sum_{I}^{n} w(x - x_I)[p^T(x_I) a(x) - u_I]^2 \qquad (2.7)$$

where: $w(x - x_I)$ is a weight function associated with node I.

The design or selection of the ”proper” weighted function is very important for the ”quality” of the approximations made [8, 17, 18].

The design of the weighting function should use the following principles:

- the value of the weighted function needs to be ≥ 0
- the weighted function must be continuous
- with the increasing distance from node x_I the function value should rapidly decrease (*the area in which the weight function takes positive values is also called a carrier*)
- the weight function must be at least the class of a shape function (i.e. if the shape function is a class of C_2, then the weight function also must be at least class C_2).

The above rules can be written formally:

1. $w(x - x_I) > 0$ inside the weighted function suport,
2. $w(x - x_I) = 0$ outside the weighted function suport,
3. $\int_{\Omega} w(x - x_I) d\Omega = 1$,
4. $w(|x - x_I|)$ is monotonically decreasing.

In the considered approximation, in the given point x, the coefficients $a(x)$ should be chosen in such a way as to minimize the weighted residual. The minimization condition can be written as:

$$\frac{\partial J}{\partial a} = 0 \qquad (2.8)$$

Equation (2.8) takes on the form of:

$$\sum_{I}^{n} w_I(x) p(x_I) p^T(x_I) a(x) = \sum_{I}^{n} w_I(x) p(x_I) u_I \qquad (2.9)$$

in the matrix form:

$$A(x) a(x) = B(x) U_S \qquad (2.10)$$

where:

$$A(x) = \sum_{I}^{n} w_I(x) p(x_I) p^T(x_I)$$

$$w_I(x) = w(x - x_I)$$

$$B(x) = [B_1, B_2, B_3, \dots, B_n]$$

$$B_I = w_I(x) p(x_I)$$

$$U_S = \{u_1, u_2, u_3, \dots, u_n\}^T$$

By solving equation (2.10) relative to $a(x)$ we obtain:

$$a(x) = A^{-1}(x) B(x) U_S \qquad (2.11)$$

$$u^h(x) = p^T(x) A^{-1}(x) B(x) U_S \qquad (2.12)$$

From (2.1) and (2.12) the approximation function shape can be determined using the method of moving least squares (MLS):

$$\Phi(x) = p^T(x) A^{-1}(x) B(x) \qquad (2.13)$$

or the shape function Φ_I linked with node I in point x:

$$\Phi_I(x) = p^T(x)A^{-1}(x)w_I(x)p(x_I) \quad (2.14)$$

A special case of a polynomial base is the zero-dimensional base, i.e. $p(x)=\{1\}$ for which the shape function of the form takes on the following form:

$$\Phi_I^0(x) = \frac{w_I(x)}{\sum_I^n w_I(x)} \quad (2.15)$$

3. Exemplary Weighted Functions

The most commonly used weighted functions are [2, 8, 16, 17, 18, 19, 21]:

– exponential weight function:

$$w(r) = \begin{cases} \dfrac{e^{-(r)^{2k}} - e^{-(d\ /cs)^{2k}}}{(1-e^{-(d/cs)^{2k}})} & r \le 1 \\ 0 & r > 1 \end{cases} \quad (3.1)$$

cs – constant determining the "shape" of the function ($cs>0$)

– cubic weighted function

$$w(r) = \begin{cases} \dfrac{2}{3} - 4r^2 + 4r^3 & r \le \dfrac{1}{2} \\ \dfrac{4}{3} - 4r + 4r^2 - \dfrac{4}{3}r^3 & \dfrac{1}{2} < r \le 1 \\ 0 & r > 1 \end{cases} \quad (3.2)$$

– square weighted function

$$w(r) = \begin{cases} 1 - 6r^2 + 8r^3 - 3r^4 & r \le 1 \\ 0 & r > 1 \end{cases} \quad (3.3)$$

where:

$$r = \frac{d_I}{d} \qquad d_I = |x - x_I| \quad (3.4)$$

d – is the size of the surroundings around the node.

In the case of two or three dimensions the following are usually used:

- circular surrounding:

$$w(x - x_i) = w\left(\frac{|x_i - x|}{d_I}\right) \quad (3.5)$$

- or rectangular surrounding:

$$w(x - x_i) = w\left(\frac{|x_i - x|}{d_I^x}\right) w\left(\frac{|y_I - y|}{d_I^y}\right) \quad (3.6)$$

Since the size of a node's surroundings depends on the adopted deployment of nodes and the size of the problem, it is practical to introduce a dimensionless indicator *rmax* [8, 18, 20] that describes the size of the surroundings around the nodes. The *rmax* indicator can be defined as a number (greater than 1) defined by formula:

$$r\max = \frac{d}{d_{\max}} \quad (3.7)$$

where:
d – surrounding size
d_{max} – maximum distance between neighbouring nodes.

4. The Solution to the Meshless Diffusion Equation Using the Petrov-Galerkin Method

In [24] the solution to the diffusion equation was presented from a meshless finite source using the Petrov-Galerkin method.

The one-dimensional diffusion problem in interval [a,b] was solved, described by equation:

$$u_t - cu_{xx} - f = 0 \quad (4.1)$$

with the following boundary and initial conditions:

$$u(a,t) = u_a \quad u(b,t) = u_b \quad u(x,0) = u_1 \quad (4.2)$$

where:
u – solution,
c – diffusion constant,
f – function dependent on the medium.

The approximation function was used in the solution with separated variables in the form of:

$$u^h(x,t) = \sum_{I \in S} \Phi_I(t) u_I(x) \quad (4.3)$$

By solving the equation using the Petrov-Galerkin method for the spatial variable and using the Crank-Nicolson method for the time variable the system of equations was obtained, which should be solved in the following time steps.

$$Au(t + \Delta t) = Bu(t) + f \quad (4.4)$$

where matrixes A, B and f are determined in accordance with the following formulas [24]:

$$A_{ij} = \begin{cases} \phi_j(x_i) & \text{for } x_j \text{ belonging to the surroundings of the node boundaries} \\ 2C_{ij} + \Delta t K_{ij} & \text{for } x_j \text{ belonging to the surroundings of the inside nodes} \\ 0 & \text{for the remaining nodes} \end{cases}$$

$$B_{ij} = \begin{cases} 0 & \text{for } x_j \text{ belonging to the surroundings of the node boundaries} \\ 2C_{ij} - \Delta t K_{ij} & \text{for } x_j \text{ belonging to the surroundings of the inside nodes} \\ 0 & \text{for the remaining nodes} \end{cases}$$

$$f_i = \begin{cases} u_a & \text{for } i = 1 \\ u_b & \text{for } i = n \\ 2\Delta t f_i & \text{for the remaining } i \end{cases}$$

$$K_{ij} = c\left[\int_{x_{QiL}}^{x_{QiR}} \phi_j'(x)\omega_i' dx - \phi_j'(x)\omega_i \Big|_{x_{QiL}}^{x_{QiR}}\right] \quad (4.4)$$

$$C_{ij} = \int_{x_{QiL}}^{x_{QiR}} \phi_j(x)\omega_i dx \quad (4.5)$$

$$F_i = \int_{x_{QiL}}^{x_{QiR}} f\omega_i dx \quad (4.6)$$

where:

ϕ_j – shape function linked with node j

ϕ_j' – shape function derivative linked with node j

ω_i – weighted function linked with node i

ω_i' – weighted function derivative linked with node i

x_{QiR} – right limit of the surroundings of node i

x_{QiL} – left limit of the surroundings of node i.

5. Presenting the Obtained Results

The simulation numerical algorithm of the diffusion process was implemented in FORTRAN [26].

The following equation was resolved:

$$u_t - cu_{xx} = 0 \quad (5.1)$$

with boundary and initial limitations:

$$u(a,t) = 0 \quad u(b,t) = 0 \quad u(x,0) = u_1(x) \quad (5.2)$$

Results were compared with the analytical solution calculated according to equation [31]:

$$u(x,t) = \frac{2}{b-a}\int_a^b \sum_{n=1}^{\infty} e^{-\left(\frac{n\pi}{b-1}\right)ct} \sin\left(\frac{n\pi}{b-a}x\right)\sin\left(\frac{n\pi}{b-a}\tau\right)u_1(\tau)d\tau \quad (5.3)$$

Calculations were done for the following boundary and initial limitations:

$a = 0$ [m], $b = 10$ [m]

$$u_1(x) = \begin{cases} \frac{1}{2}(1+\tanh(\alpha(x-x_1)))\ [\frac{1}{m}] & dla\ x < \frac{a+b}{2} \\ \frac{1}{2}(1-\tanh(\alpha(x-x_2)))\ [\frac{1}{m}] & dla\ x \geq \frac{a+b}{2} \end{cases} \quad (5.4)$$

where:

$$\alpha = \frac{N-1}{b-a}$$

$$x_1 = a + 0{,}3*(b-a)$$

$$x_2 = a + 0{,}7*(b-a)$$

N – number of nodes in interval $[a,b]$.

Function u_1 appears as in figure 1.

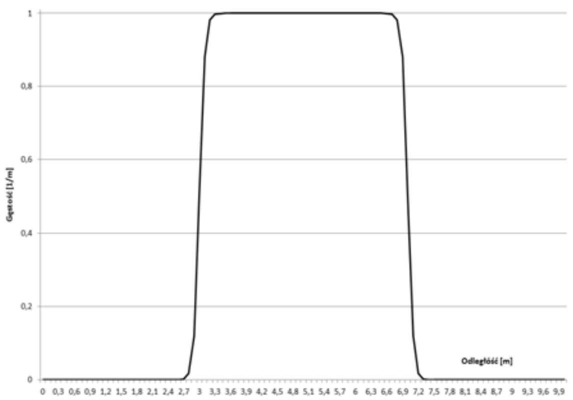

Fig. 1. Shape of function u_1

The following number of nodes were accepted N = [60, 80, 100, 120, 140, 160, 180, 200] and their equal distribution. The distance between nodes is presented in table 1.

Tab. 1. Distance between nodes depending on their number

Number of nodes	Distance between nodes [m]
60	0.169
80	0.127
100	0.101
120	0.084
140	0.072
160	0.063
180	0.056
200	0.050

The weighted function and weighted function derivative graphs used in the simulation is presented in figure 2.

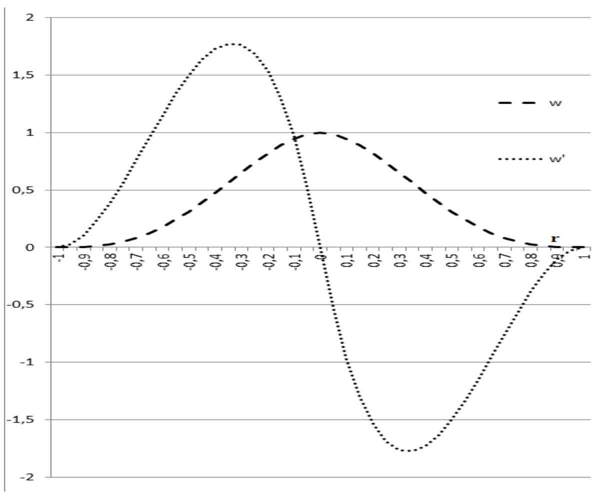

Fig. 2. Weighted square function and its derivative

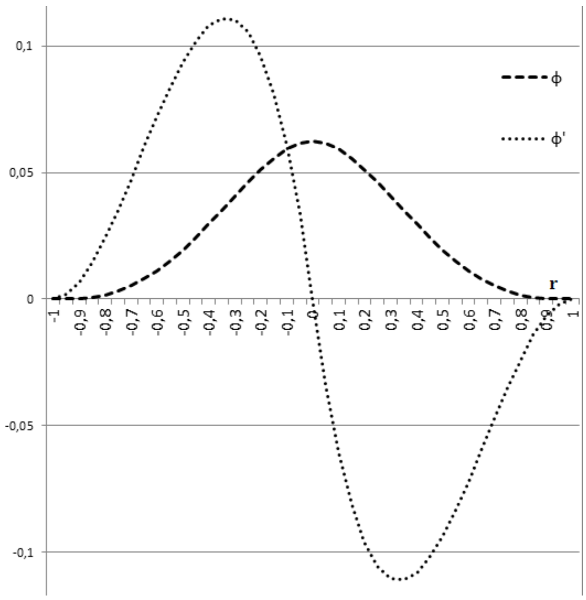

Fig. 3. Shape function and its derivative

Approximation of the moving least squares (MLS) with a zero-dimensional base. The shape function and shape function derivative graph is presented in figure 3. The following calculation algorithm was accepted:

- for the assumed initial and boundary limitations calculate the analytical solution in interval [*a*,*b*] for *n* nodes at time of *t*
- start the meshless algorithm
  - for each step calculate the mean square error of the results obtained in the nodes for meshless and analytical solutions
  - complete the calculation if the error obtained in the next step is larger than the error from the previous step
  - assume the outcome as the results obtained in the previous step
- determine the parameters of the obtained solution:
  - time after which the algorithm obtained the solution – as the product of the number of steps and time step
  - sum of values of the absolute difference in the result and analytical result
  - mean square error of the difference in the result and analytical result.
- record the obtained results.

Research was performed on following the algorithm parameters

- the accuracy of the results,
- range of variation of the algorithm parameters, such as time step *dt* and the *rmax* coefficient for which solutions have physical form,
- behaviour of the algorithm for solutions pursuing the constant value.

In the first step an analysis was conducted for the following parameter values:

- diffusion coefficient 0.05, time *t* = 50s
- time step *dt* = [1s; 1.5s; 2.0s; 2.5s; 3.0s; 3.5s; 4.0s; 4.5s]
- coefficient *rmax* = [1.1; 1.4; 1.7, 2.0; 2.3].

Variation range of the *rmax* coefficient was accepted on the basis of literature data [8, 18, 20].

Calculations were done on an Intel® Core™2 Quad Q8300 2,50GHz 6GB RAM computer with Windows 7 Professional SP1 64bit.

Based on the analysis of the achieved results it was ascertained that:

- for the selected parameters time shown by the implemented algorithm, for the smallest mean square error compared to analytical solutions, differs from the time of the analytical solution
- mean square error:
  - increases along with the increase of nodes
  - increases along with the increase of the *rmax* coefficient.

These results are visible in figures 4 and 5.

Since for the selected parameters the time to obtain the analytical result by the implemented meshless algorithm, depending on the number of nodes and the *rmax* coefficient, seems to be a linear curve, calculations were carried out to determine these dependencies.

The following values were accepted:

- diffusion coefficient 0.05
- time *t* = [10s,25s,40s,55s,70s]
- time step *dt* = [1; 1.5; 2.0; 2.5; 3.0; 3.5; 4.0; 4.5]
- coefficient *rmax* = [1.3; 1.8; 2.3]

These results are visible in figures 6, 7 and 8.

Based on the obtained numbers the time dependencies of the implemented algorithm time specified, the results are included in table 2.

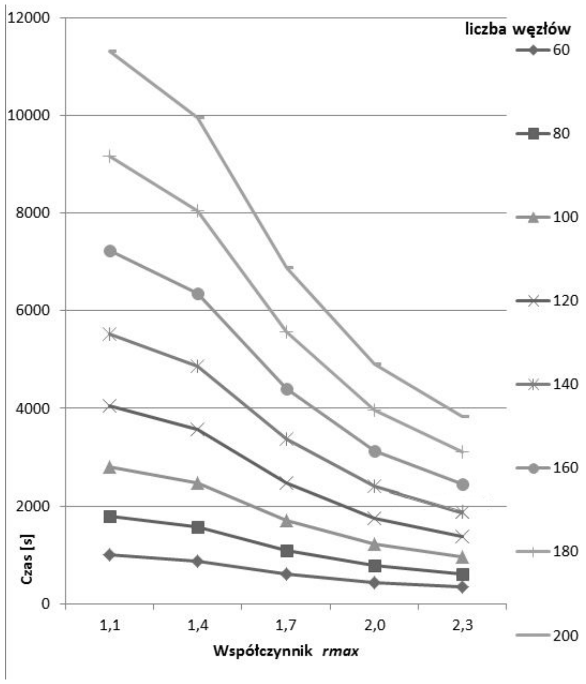

Fig. 4. The time at which the meshless algorithm achieved an analytical solution depending on the *rmax* coefficient and number of nodes

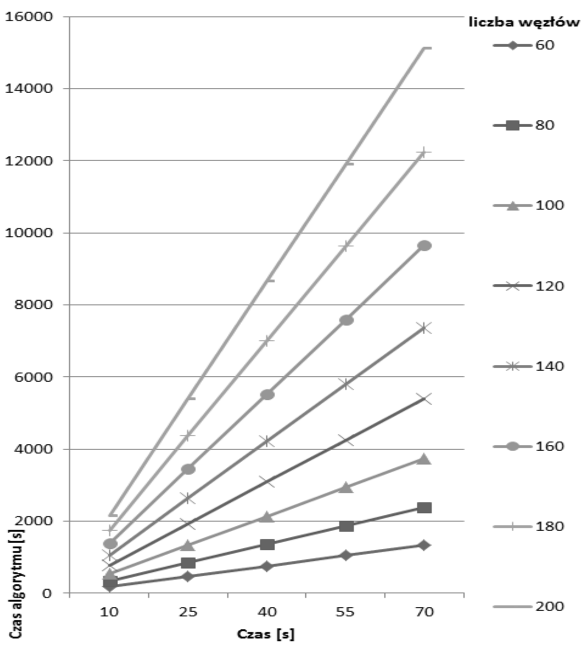

Fig. 6. The time after which the meshless algorithm achieved an analytical result depending on the number of nodes for *rmax* = 1.3 and diffusion coefficient = 0.05

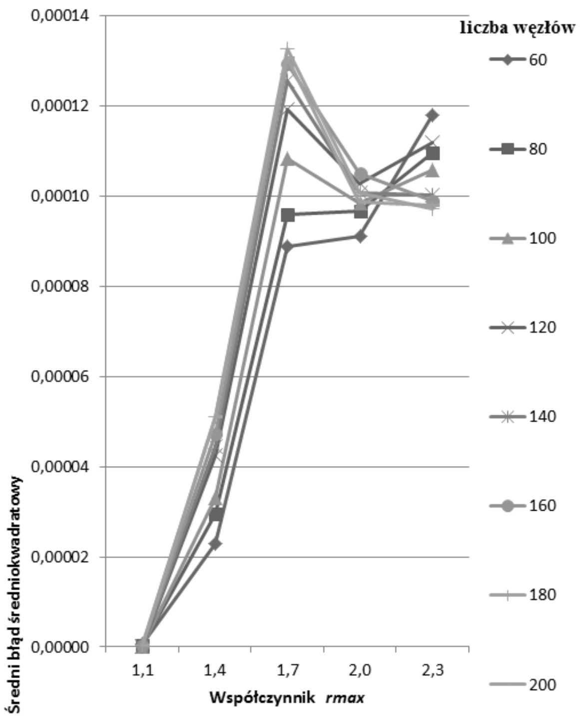

Fig. 5. The average mean square error dependency on the *rmax* coefficient and the number of nodes

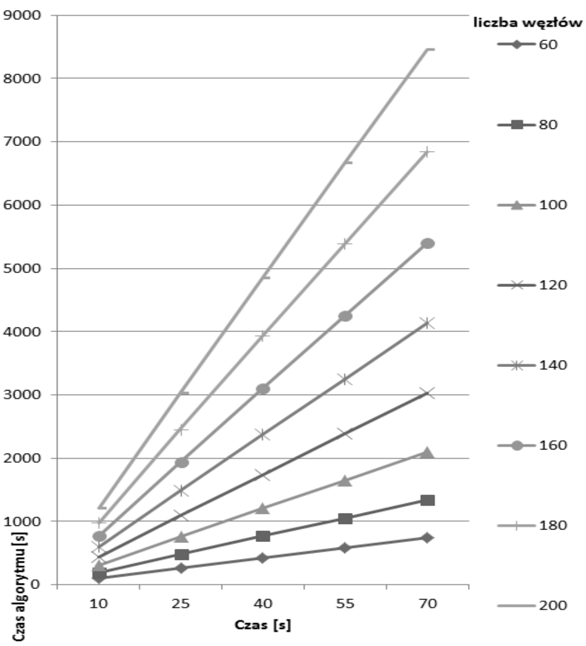

Fig. 7. The time after which the meshless algorithm achieved an analytical result depending on the number of nodes for *rmax* = 1.8 and diffusion coefficient = 0.05

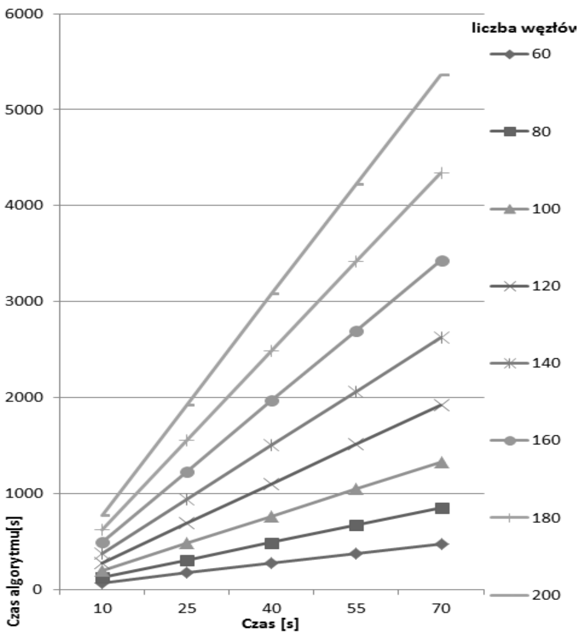

Fig. 8. The time after which the meshless algorithm achieved an analytical result depending on the number of nodes for *rmax* = 2.3 and diffusion coefficient = 0.05

Dependencies were used to determine the accuracy of the implemented algorithm for the following parameters:

- diffusion coefficient 0.9:
  - time *t* = [1s,2s,3s,4s,5s]
  - time step *dt* = [1; 1.5; 2.0; 2.5; 3.0]
  - coefficient *rmax* = [1.3; 2.3]
- diffusion coefficient 0.3:
  - time *t* = [1s,11s,21s,31s,41s]
  - time step *dt* = [1; 1.5; 2.0; 2.5; 3.0]
  - coefficient *rmax* = [1.3; 1.8; 2.3]
- diffusion coefficient 0.2:
  - time *t* = [1s,11s,21s,31s,41s]
  - time step *dt* = [1; 1.5; 2.0; 2.5; 3.0]
  - coefficient *rmax* = [1.3; 1.8; 2.3]
- diffusion coefficient 0,1:
  - time *t* = [1s,11s,21s,31s,41s]
  - time step *dt* = [1; 1.5; 2.0; 2.5; 3.0]
  - coefficient *rmax* = [1.3; 1.8; 2.3]
- diffusion coefficient 0,05:
  - time *t* = [10s,110s,210s,310s,410s]
  - time step *dt* = [1; 1.5; 2.0; 2.5; 3.0]
  - coefficient *rmax* = [1.3; 1.8; 2.3]
- diffusion coefficient 0,005:
  - time *t* = [10s,410s,810s,1210s,1610s]
  - time step *dt* = [1; 1.5; 2.0; 2.5; 3.0]
  - coefficient *rmax* = [1.3; 1.8; 2.3].

Tab. 2. Time dependencies of the implemented algorithm

rmax	No. of Nodes	MLPG Time Dependency on Real-time
1.3	60	MLPG Time = 18.996*time + 2.767
	80	MLPG Time = 34.081*time + 2.592
	100	MLPG Time = 53.523*time + 2.617
	120	MLPG Time = 77.311*time + 3.212
	140	MLPG Time = 105.467*time + 3.667
	160	MLPG Time = 138.043*time + 3.373
	180	MLPG Time = 174.978*time + 3.454
	200	MLPG Time = 216.289*time + 3.102
1.8	60	MLPG Time = 10.603*time + 4.029
	80	MLPG Time = 19.09*time + 3.202
	100	MLPG Time = 29.955*time + 3.9
	120	MLPG Time = 43.286*time + 4.025
	140	MLPG Time = 59.1*time + 3.437
	160	MLPG Time = 77.319*time + 4.160
	180	MLPG Time = 97.992*time + 4.367
	200	MLPG Time = 121.123*time + 4.367
2.3	60	MLPG Time = 6.727*time + 3.462
	80	MLPG Time = 12.106*time + 3.156
	100	MLPG Time = 19.013*time + 3.129
	120	MLPG Time = 27.449*time + 3.498
	140	MLPG Time = 37.478*time + 3.329
	160	MLPG Time = 49.019*time + 3.848
	180	MLPG Time = 62.122*time + 3.923
	200	MLPG Time = 76.747*time + 4.514

Maximum and minimum value of the mean square error are presented in table 3.

Tab. 3. Maximum and minimum mean square error depending on the accepted simulation parameters

Diffusion Coefficient	*rmax*	Mean Square Error	
		Maximum	Minimum
0.9	1.3	0.0066	0.000017
	2.3	0.1163	0.000223
0.3	1.3	0.0024	0.000014
	1.8	0.0187	0.000128
	2.3	0.0425	0.000186
0.2	1.3	0.0016	0.0000056
	1.8	0.0123	0.0000834
	2.3	0.0274	0.0000756
0.1	1.3	0.00077	0.00000097
	1.8	0.00584	0.00001182
	2.3	0.01295	0.00000942
0.05	1.3	0.00016	0.00000024
	1.8	0.00058	0.00000515
	2.3	0.00073	0.00000813
0.005	1.3	0.0121	0.00000035
	1.8	0.0087	0.00000059
	2.3	0.0084	0.00000099

6. Conclusion

1. Using the algorithm for the numerical simulation of variable processes in time requires determining the range of input parameters for which the results have a physical meaning.
2. The meshless implemented algorithm can be successfully applied to time-dependent problems, aiming at a stationary solution, if such exists.
3. The meshless implemented algorithm obtains correct results for the tested diffusion coefficient range.
4. For each task, it is necessary to make a selection of a coefficient determining the size of the surrounding around the node (*rmax*) and the time step (*dt*).
5. Analyses show that the range for *rmax* = [1.3-2.3] is admissible.
6. In case of a coefficient value *rmax* > 2.3 the algorithm is divergent.
7. The *dt* time step algorithm should be chosen depending on the size of the diffusion coefficient and simulation time.
8. For small values of the diffusion coefficient it can take higher values, while in the case of a large diffusion coefficient the time step should be small.
9. The implemented meshless algorithm behaves correctly for solutions pursuing a constant value.

7. Bibliography

[1] S.N. Atluri, S. Shen, ”The meshless local Petrov–Galerkin (MLPG) method: a simple and less-costly alternative to the finite element and boundary element methods”, *Comput. Model Eng. Sci.* 3, 11–51 (2002).

[2] S.N. Atluri, T. Zhu, ”A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics”, *Comput. Mech.* 22, 117–127 (1998).

[3] Benbouza, N. Louai, F.Z. Nait-Said, N., Application of Meshless Petrov Galerkin (MLPG) method in electromagnetic using Radial Basis Functions, PEMD 2008. 4th IET Conference, 650− 655 (2008).

[4] Benoudjit, A., F.Z. Louai, N. Benbouza and S. Drid, 2007. Parametric analysis by the meshless local petrov galerkin (MLPG) approach applied to electromagnetic problems, *Int. J. Electrical Power Eng.*, 1: 138−145.

[5] Lu Huifen Jiang Hao, ”Application of meshless local Petrov-Galerkin methods in electromagnetics”, Electrical Machines and Systems, ICEMS 2003, Sixth International Conference, vol.2, 798−801 (2003).

[6] S. N. Atluri, H.-G. Kim, J. Y. Cho, ”A critical assessment of the truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) methods”, *Computational Mechanics*, 24, 348-372 (1999).

[7] Bangti Jin, Yao Zheng, ”A meshless method for some inverse problems associated with the Helmholtz equation”, *Comput. Methods Appl. Mech. Engrg*. 195, 2270–2288 (2006).

[8] J. Dolbow, T. Belytschko, ”An introduction to programming the meshless Element Free Galerkin method”, *Archives of Computational Methods in Engineering*, 5, 207−241 (1998).

[9] Gleber Nelson Marques, Jose Marcio Machado, Sergio Luis Lopes Verardi, Stephan Stephany, Airam Jonatas Preto, ”Interpolating EFGM for computing continuous and discontinuous electromagnetic fields”, COMPEL: *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 26, 1411–1438 (2007).

[10] S. L. Ho, Shiyou Yang, H. C. Wong, and Guangzheng Ni, ”A Meshless Collocation Method Based on Radial Basis Functions and Wavelets”, *IEEE Transactions On Magnetics*, 40, 1021−1022 (2004).

[11] Shiyou Yang, Guangzheng Ni, J. R. Cardoso, S. L. Ho, and J. M. Machado, A Combined Wavelet-Element Free Galerkin Method for Numerical Calculations of Electromagnetic Fields, IEEE TRANSACTIONS ON MAGNETICS, 39, 1413−1416 (2003).

[12] S. L. Ho, Shiyou Yang, and H. C. Wong, ”Weak Formulation of Finite Element Method Using Wavelet Basis Functions”, IEEE TRANSACTIONS ON MAGNETICS, 37, 3203−3207 (2001).

[13] Y. Markcha1, ”Some Meshless Methods For Electromagnetic Field Computations”, IEEE TRANSACTIONS ON MAGNETICS, Vol. 34, 3351−3354 (1998).

[14] S. L. Ho, S. Yang, J. M. Machado, and H. C. Wong, ”Application of a Meshless Method in Electromagnetics”, IEEE TRANSACTIONS ON MAGNETICS, Vol. 37, 3198−3202 (2001).

[15] Ph. Bouillarda, S. Suleaubb, ”Element-Free Galerkin solutions for Helmholtz problems: fomulation and numerical assessment of the pollution effect”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 162, 317−335 (1998).

[16] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, ”Meshless methods: An overview and recent developments”, *Computer Methods in Applied Mechanics and Engineering*, 139, 3-47 (1996).

[17] Vinh Phu Nguyena, Timon Rabczukb, Stéphane Bordasc, Marc Duflotd, ”Meshless methods: A review and computer implementation aspects”, *Mathematics and Computers in Simulation*, Vol. 79, 763−813 (2008).

[18] T. Belytschko, Y.Y. Lu, L. Gu, ”Element-Free Galerkin Method”, *International Journal for Numerical Methods Ingineering*, 37, 229−256 (1994),

[19] Stephen Beissel, Ted Belytschko, ”Nodal integration of the element-free Galerkin method”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 139, 49−74 (1996).

[20] Vlatko Cingoski, Naoki Miyamoto and Hideo Yamashita, ”Element-Free Galerkin Method for Electromagnetic Field Computations”, *IEEE Transactions on Magnetics*, Vol. 34, No 5, 3236−3239, SEPTEMBER 1998.

[21] J. Dolbow, T. Belytschko, ”Numerical integration of the Galerkin weak form in meshless methods”, *Computational Mechanics*, 23, 219−230 (1999).

[22] Shaofan Li, Dong Qian, Wing Kam Liu, and Ted Belytschko, ”A meshless contact-detection algorithm”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, Issues 24−25, 3271−3292, March 2001.

[23] Liviu Marin, ”A meshless method for the numerical solution of the Cauchy problem associated with three-dimensional Helmholtz-type equations”, *Applied Mathematics and Computation*, 165, 355–374 (2005).

[24] Marjan Sterk , Borut Robic , and Roman Trobec, Mesh Free Method Applied to the Diffusion Equation, Parallel Numerics 57−66 (2005).

[25] J. Gawinecki Z. Domański, Matematyka. Równania różniczkowe cząstkowe i metody ich rozwiązywania. Częśc II. Równiania różniczkowe i cząstkowe II-go rzędu i metody ich rozwiązywania. Skrypt WAT wewn. 2490/97, Sygn. S-54433

[26] http://pracownicy.uwm.edu.pl/wojsob/projekty-edi.htm

[27] R.A. Gingold, J.J. Monaghan, ”Smoothed particle hydrodynamics: theory and application to non-spherical stars”, *Monthly Notices R. Astron. Soc.* 181, 375–389 (1977).

[28] L.B. Lucy, ”A numerical approach to the testing of the fission hypothesis”, *Astron. J.,* 82, 1013–1024 (1977).

[29] L.D. Libersky, A.G. Petscheck, T.C. Carney, J.R. Hipp, F.A. Allahdadi, ”High strain Lagrangian hydrodynamics”, *J. Comput. Phys.* 109, 67–75 (1993).

Własności bezsiatkowej metody symulacji dyfuzji ze źródła skończonego

T. DAWIDOWICZ

W pracy przedstawiono analizę właściwości algorytmu numerycznego symulacji dyfuzji ze źródła skończonego opartego o bezsiatkową metodę numeryczną. Wykazano zależności czasowe symulacji oraz przedstawiono błędy symulacji. Wskazano zakres parametrów, dla których metoda uzyskuje wyniki fizyczne.

Słowa kluczowe: metoda bezsiatkowa, dyfuzja, symulacja.

Enhancing Usability of the Human-Computer Interfaces with Software Architecture

R. KASPRZYK, J. SROCZYŃSKI, K. STELMACH, P. SZADKOWSKI, K. WAWRUCH
rkasprzyk@wat.edu.pl

Institute of Computer and Information Systems
Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw

The article tells about two projects, which pay a great attention to the architectural side of a non-standard human-computer interface. The first project is THEIA, which is a touchless human-computer interaction system. The main target is to give the end-user an ability to interact with a computer only by his/her sight. By looking at areas of the screen the user triggers actions in the operating system he/she is using. The second project is OLIMPUS designed for creating and integrating modern user interfaces supporting any non- standard interaction sources. The project aims at simplifying a development process of interfaces collaborating simultaneously with many different sources of interaction and their dynamical exchanging as well.

Keywords: interaction, interface, architecture.

1. Introduction

Nowadays there are a lot of possibilities of the interaction with a computer – good quality webcams make it possible to create an image analysis based on interaction, the growing power of CPUs allows to perform speech recognition in real-time and much more. But, in fact, any attempt to create such an advanced solution is by now uneconomical. The problem is that most of the revolutionary interface solutions, to show their potential, need a dedicated implementation in the end-user software. Doing it in only a few applications will change nothing. Doing it with most of popular software is too expensive way.

In this article two projects are described, which are related to the human computer interaction area. The difference is, that they both are trying to solve the problem of practical usage of them with their architecture. The first one – THEIA – by delivering a relatively easy way to connect any application to its interaction engine or extend its functionality. The second one – OLIMPUS – by creating a platform for integrating custom interfaces with a prepared consumer application, without any need of any changes in the used components and without any additional costs of such integration.

2. The THEIA project

The THEIA (stands for: THEIA – The Handy Eye Interaction Adapter) system was planned as a universal interaction software tool for touchless interaction with the personal computer. The communication with a standard PC is based on an eye position recognition and further a user's gaze point by analyzing the image obtained from a regular web camera.

The main reason for developing such a solution is to help a lot of people with many different disabilities (for instance ”locked-in syndrome”, paresis or paralysis) use a computer in order to acquire electronic sources of knowledge, or simply contacting the world. Moreover, the THEIA system can monitor such user's activity and alarm a carer in an emergency. In addition to facilities delivery to disabled people, THEIA has also aimed at business, industry and institution sectors as well as end user facilities providing increases of interactivity with computers and bringing benefits from it. General functional requirements are presented in the Fig. 1.

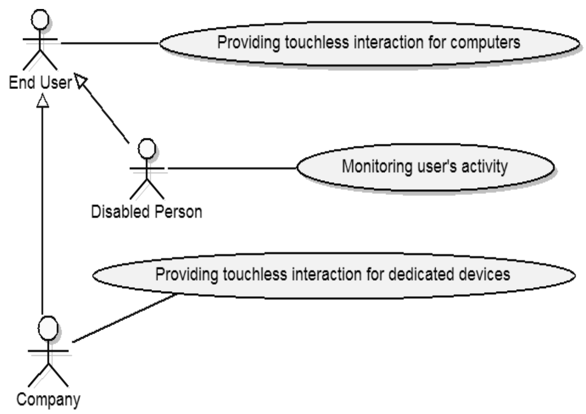

Fig. 1. Use Case Diagram for THEIA

However, making THEIA the competitive product on the market means eliminating disadvantages of existing solutions with similar capabilities, the project has also some assumptions connected with non-functional requirements, which are as follows:

- system supported by a regular web camera or a better one
- computing a user's gaze point only based on image data, without any face markers, infrared radiation etc.
- optimizing performance's issues as much as possible, at least over a dozen images (about 15 FPS) obtained from a camera should be analyzed
- minimizing CPU and memory usage for the rest of the running programs.

3. THEIA's prototype outline

The idea was to design a new system steering a dedicated user interface able to handle the whole functionality of a normal operating system nowadays and meet the requirements brought by the future.

As far as the architecture is concerned, it is built around three main components:

- Core component
- Overlay Manager
- Overlays' system.

Adding a communication system mainly based on pipe technology and design a communication protocol makes the architecture very modular. The whole idea is depicted in the diagram (Fig. 2).

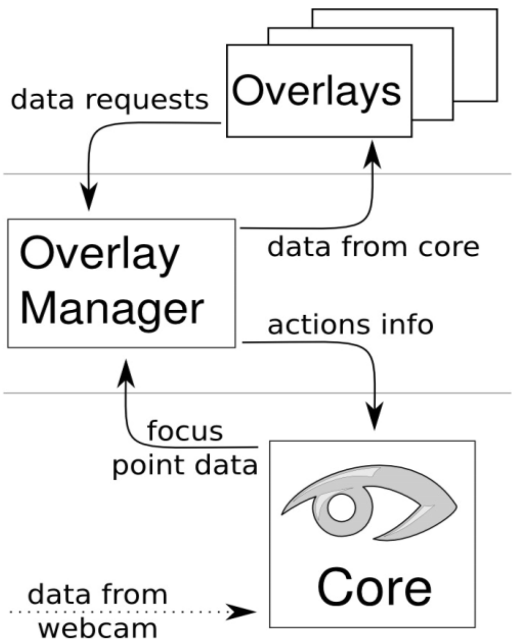

Fig. 2. THEIA architecture diagram

The Core component is responsible for obtaining image captured by the web camera and analyzing it (Fig. 3). The analysis, preceded by smoothing obtained image data, converting it to a gray scale image and equalizing its color's histogram, includes firstly user's face position recognition and secondly eyes' areas detection, using appropriate models of the face and eyes. At the next stage the eyes' centers are being calculated in the following steps:

- finding eyes' pupils (mainly black color) inside eyes' areas boundary (mainly white to gray color)
- enlarging found eyes' pupils' images using an extra algorithm, developed at the ITA institute, Military University of Technology, sharpening the edges of objects visible in the image (more information is included in [15])
- detecting edges with Canny algorithm (an idea of edges detecting can be found in [5]) and further Hough Transform (more in publication [6]) in the enlarged eyes' pupils' images
- computing pupils' centers using obtained circles from Hough Transform.

The pupils' centers data are the basis for calculating a user's gaze point directed at the monitor screen. In addition, once face is found, it is constantly being tracked by THEIA Core and the whole analysis is being repeated. Whenever the user's face is lost, THEIA Core will automatically search for it once again. All these steps are performed in such a way that covers most non-functional requirements.

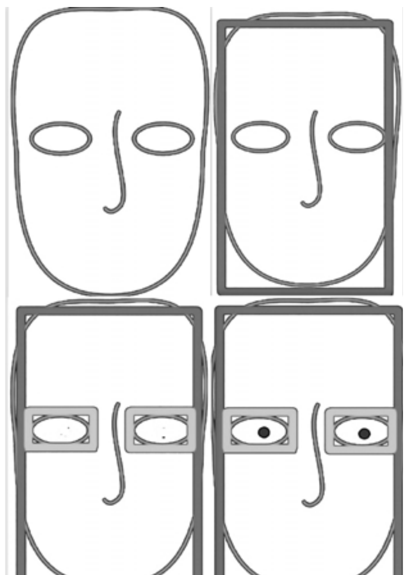

Fig. 3. THEIA Core image analyzing steps

All computed data is passed through to the Overlay Manager, which handles the received data and controls the behavior of all components by triggering suitable actions. Moreover, it retrieves all requests sent from overlays as well as sends its own messages to both Overlays and the Core component. The whole process is reduced to decode the obtained messages, identify the requested action, encode a new message and send it to its receiver. This method is strictly connected with the communication protocol implying sending only formatted text messages according to the following pattern:

```
FUNCITON_NAME,param1,..,paramN
```

The parameters' list can vary depending on the specific function name. Some examples have been given below:

```
COORD,X,Y
ACTIVEAREA,AREA_ID
CALIBRATION
```

Furthermore, the design protocol does not assume elements connected with identifying sender and receiver of the message as far as the stack mechanism supports management of the overlays. It assures that only one overlay can be active at the same time and as a result has exclusive right to sending and receiving messages. This mechanism provides a sort of automatic feedback messaging. On the other hand, the communication between the Core component and Overlay Manager is uniquely identified by the separate pipe connection. There can be only one such connection simultaneously. Such a solution allowed to fairly well collaborate with the used pipe technology and make it easily extensible with new components, like overlays.

Each Overlay is independently of one another working module providing special set of functionalities as are available in the operating systems. However, they can collaborate with one another only by communicating with Overlay Manager, giving a possibility to combine different functions to cover almost all, both basic and more sophisticated, areas of using a computer. Nevertheless, Overlays' system possesses some restrictions, connected with a computing precision of the gaze point, because of poor camera quality. All this has forced to design a special grid in which every single grid's field corresponds to a specific activity triggering implemented functionality. The grid could be any size but still it never consists of number of rows and columns grater than several ones.

The entire project was developed using Microsoft technologies. Every component was implemented in C# under .NET Framework 3.5. The core component was mainly based on EmguCV library (please visit home page to learn more [4]), which is a set of a different kind of image operations based on OpenCV project adapted for C# (more in [2], [3]). As it has been mentioned before, the models of the face and eyes for their detection are stored in several separate XML files as a part of EmguCV library. As far as graphical user interfaces (GUI) for each overlays are concerned it was designed as WPF application, including some 3D interactive components, if necessary, using 3DTools DLL library.

Having regard to some system's limitations and used solutions, that should be improved, THEIA will allow controlling computer only by the gaze. Such an opportunity has been noticed and positively evaluated by many organizations who have become acquainted with the concept of THEIA. As a result, it has won several awards both at home and abroad.

4. The revision of the objectives

The prototype of the THEIA system was a good basis for further analysis. It allowed to check, how much a created solution can be useful. The evaluation of the practical usefulness of the system, based on the market review and real world tests, have given some interesting observations, and changed some targets of the project.

The fact, which seems to be certain after some research, is that there appeared other systems in development, which offer the same functionality as the THEIA's eye-tracking core. Basic targets for those projects are nearly similar to THEIA's, and they are progressing very well. For example there is the Cyber-Eye system[7], created in the Multimedia Systems' Department at Gdańsk University of Technology. This system is based on infra-red lighting, and offers great accuracy of sight detection. The another example is the OpenGazer system [8], which performs exactly the same activities as THEIA's core module. It has a support of the Samsung company, the Gatsby Foundation, and the European Commission (as it is a part of AEGIS program). Both systems are in an advanced stage, and practically are more accurate and universal than THEIA's mechanisms. In that case any further development of the eye-tracking module of THEIA should be considered as pointless.

On the other hand, THEIA's extensible architecture performed very well in practice. It was its triumph card, because most of the innovative interfaces have not developed any suitable way to implement them in the real world, like THEIA's overlays' system or communication protocol. The chance to bind any application with THEIA's core with a small effort, or freedom with extending its functionality became a crucial advantage of this project.

Because of this an idea has emerged to try not only to create another way of communication with the computer, but mainly focus on what can be built from already existing human-computer interface devices and software solutions, and how to bind them to maximize the benefits for the end-user, but do it in a cheap and easy way. That is how the OLIMPUS project was initiated, the successor of THEIA (OLIMPUS stands for Overall, Light, Interactive, Multi-Purpose User-interface System).

5. The OLIMPUS overview

The OLIMPUS project aims to create a software platform which delivers to the end-user an easy to use but robust tool to bind the available ways of interaction with application software. The platform should give an ability to define what types of interaction the user wants to use with the applications, and for which actions in software applications each way of interaction is responsible. Those responsibilities should be able to dynamically change or expand on users' demands.

Usually, in most consumer applications, the functionalities are able to be triggered by certain actions. These actions are strictly connected to certain interaction events (for example a key shortcut is related to an interaction with the keyboard only). The general scheme of such a relation is illustrated in Fig. 4. That makes all applications bind to their dedicated interaction events, and makes them immune to the unusual interaction methods, as far as these interaction methods are no trying to pretend that events.

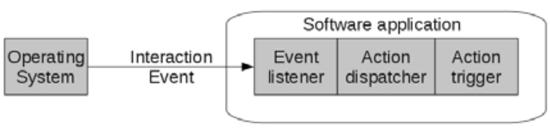

Fig. 4. Standard action triggering (based on [9])

The basic idea for the OLIMPUS project is to detach the interaction method from an action in software, and create an ability to freely match different interaction methods with the needed functionality. To achieve this there is a need to isolate an event listener from the application (Fig. 5), and through this make it switchable with another. That moves the interaction dependence from the application to the dedicated event listener. Then, with changing the listener, a change of the interaction method with the application takes place.

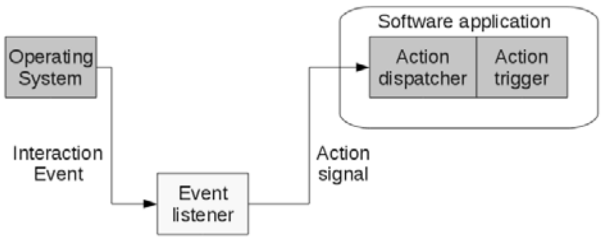

Fig. 5. Proposed action triggering

The OLIMPUS structure is based on this idea. It consists of three main components:

- interaction plugins, which are the mentioned above event listeners, isolated as a standalone applications capable of inter-process communication
- interface plugins, which are the mentioned above software applications, capable of receiving action signals for triggering actions
- the core component, which is a software tool responsible for connecting the interaction plugins with the interface plugins in a desired manner.

The interaction plugins are responsible for receiving data from an interaction input device connected to the computer, generate some action signals based on them, and send these signals to the core component. The core component is able to find out which interface plugin should use these signals, and deliver them to it. This manner of performing action triggering in applications makes an opportunity to freely select the interaction method to suite the user's demands, or power in the interface plugin with many interaction methods simultaneously (as in the Fig. 6).

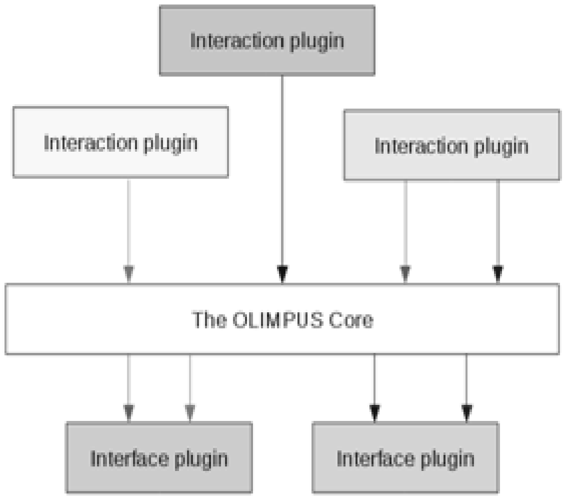

Fig. 6. The OLIMPUS structure idea

Of course there is a need to prepare the applications to act as the interface plugin, or to create the interaction plugin, which is utilizing some interaction method. Because of that a large-scale implementation of this solution needs a solid support from applications' developers and interaction methods creators. But preparing software to work with OLIMPUS should be much easier and cheaper than altering applications every time, when new suitable interaction method will show up, and gives great possibilities in a customization of the interaction without any further costs.

6. The OLIMPUS core

The OLIMPUS core is the main part of the solution. Its design is still at the development stage, but already most of its fundamental functionality is designed and implemented. The whole platform is created with Java technology [10], because of its great set of libraries and extensions, suitable for such applications. The specific technologies will be described with related functions of the application. The core is responsible for critical activities of the whole platform, such as:

- executing and initializing all of available plugins
- performing the configuration of plugins, including a universal calibration GUI for interaction plugins, if needed
- creating a scheme of connections between plugins
- routing interaction signals between plugins accordingly with the created scheme.

Those are the main areas of the OLIMPUS' core functionality, and they are going to be explained a bit in this part.

As mentioned earlier, it was decided, that plugins should be delivered to the platform, as stand-alone applications, capable of inter-process communication with the core. It makes adding new plugins to the system much easier, because there is no need to change anything in the core itself. There is only a need to give it information which plugin has to be connected to the core and what it is capable of. Secondly, it makes all plugins technology-independent – the only requirement is that the plugin must support the OLIMPUS' way of communication. Plugins themselfs are responsible for:

- interaction plugins – delivering interaction signals from user-interaction devices,
- interface plugins – utilizing action signals (e.g. for in-plugin actions).

As a communication technology was chosen XML-RPC [11]. It is a well defined, easy to use and widely implemented method of remote procedure calling. It uses HTTP as transport and XML as encoding. In practice it needs a dedicated, lightweight HTTP server running on each application connected to the core for receiving requests, and a technology-dependent client library to send requests.

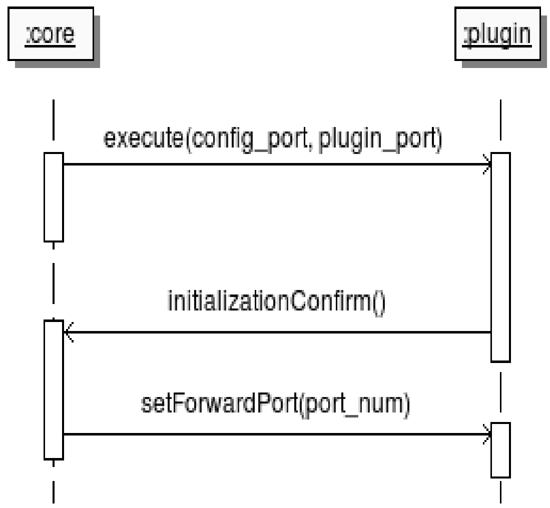

Fig. 7. Plug-in initialization sequence

The only problem with setting up links between plugins and the core is with port numbers (the IP adress is a minor problem by now, because OLIMPUS is actually aimed at work mainly within the localhost area). To solve that problem there is a fixed launch process for each of plugins, which gives the responsibility to set up all ports to the core (the process is presented in Fig. 7).

During this process, the core executes the plugin with parameters – ports for configuration communication. The first parameter of the plugin execution is the port on which the core's XML-RPC server is running. The second one is the port on which the plugin should start its own XML-RPC server. When the plugin is ready, it sends a confirmation signal to the core. Finally, the core sends to the running plugin information about "forward port", the port to which the plugin should send action signals – it will be explained later.

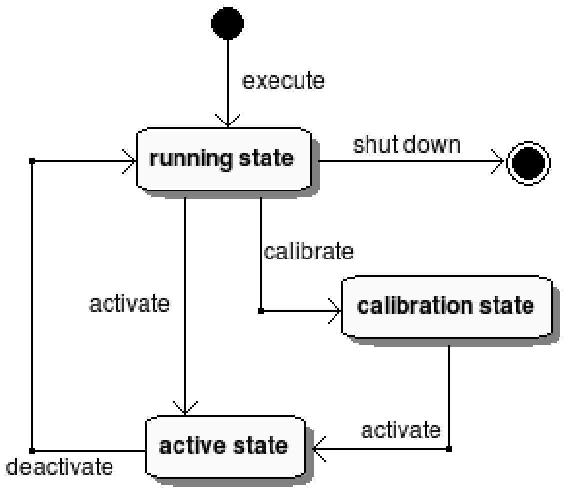

Fig. 8. States of the plug-in

When all these ports are set, it allows the plugins to receive configuration requests from the core. The plugin configuration does not include the acquisition of data about plugin capabilities – these data are stored within the extra file delivered with the plugin, which is read before the plugin execution. The configuration is performed by a specific API, and covers such areas as defining active interaction methods or controlling the state of the plugin (Fig. 8). By now, there are specified three characteristic states for plugins:

- running state – the plugin is running, but it is not sending or receiving any interaction signals to the core,
- active state – the plugin is running, and performing its interaction activities,
- calibration state – an optional state, in which the plugin, if needs to, can perform its own calibration or initialization processes.

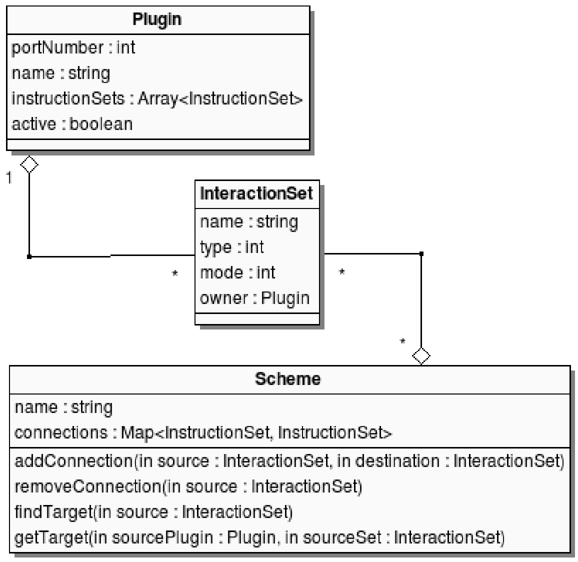

Fig. 9. Scheme structure

The next thing is a scheme of connections between plugins, which is needed, because the core must know, where to direct signals from interaction plugins. It is created before plugins' activation. The structure, which is capable of storing the whole scheme is presented in Fig. 9 as a class diagram. This structure is relatively simple, but sufficient. The plugin class stores all the data about the plugin, including a group of interaction sets, of which it is capable. An interaction set is meant as a pack of action signals, which can be generated by the plugin, grouped by a similar purpose or some other criterion (but meant to be used together). The scheme contains information to which plugin a signal from a specific interaction set should be sent, and delivers operations, which support use of these data. It is achieved by joining instruction sets within the scheme in pairs – source and target instruction set of the same interaction type. But what does it mean – source and target?

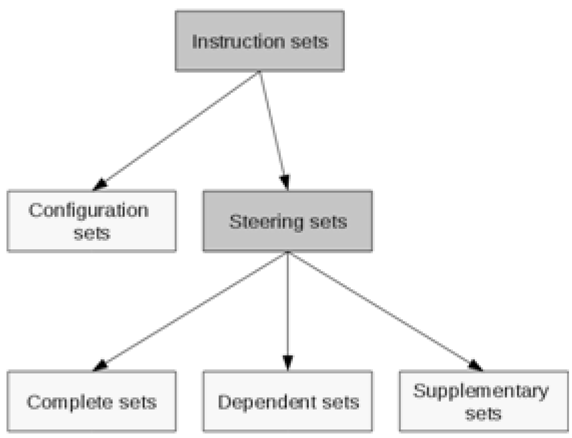

Fig. 10. Instruction sets classification

There is a need to describe instruction sets. Because an instruction set is related to some section of a specific type of human-computer interaction, there can be a huge variety of such sets. To bring some order, there has been created a classification of instruction sets. First of all, there are two types of them – sources and targets. Sources are associated with interaction plugins, and are meant to send interaction signals. Targets are meant to receive signals, and they are associated with interface plugins (like consumer applications). In addition instruction sets are grouped by their control capabilities. The classification (Fig. 10) consists of 4 groups:

- configuration sets – only used for plugin configuration, and access to core controls
- complete sets – capable of handling the interaction with computer individually
- dependent sets – unable of handling the interaction with computer individually, but it is possible with some other dependent sets
- supplementary sets – prepared only to add more functionality to interaction, insufficient to be treated as a significant way of interaction.

There is an attempt to categorize dependent sets, to get an ability to evaluate when a group of such sets could be treated as one complete set. The most promising subdivision of dependent sets is based on their interaction purposes and is presented in Fig. 11. With such categorization there is a chance, that the core will be able to automatically ensure, if the interface plugin, according to the active connection scheme, have enough interaction plugins connected to it to be fully operational for the user.

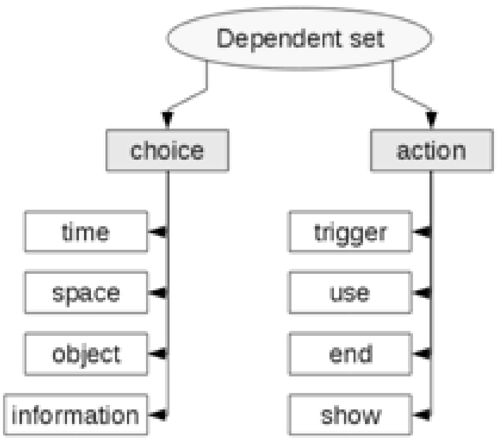

Fig. 11. Dependent instruction sets subcategorization

The connection schemes are crucial for the routing, or the so-called forwarding process. When the plugins are active, some of them (interaction plugins) are sending signals to the core, other (interface plugins) are waiting for those signals from it. The core must find out, using the connection scheme, where specific signals should be delivered, and direct them to the right interface plugin. This process is almost similar to the work performed by a standard LAN switch [12]. That is why the core sends to plugins, during the initialization process, the additional port number – a forwarding port. On this port the core is running a "raw" HTTP server (the server which does not decode requests into RPC calls, but takes them, as they are). All interaction plugins are sending interaction signals to this server. It is meant to read a request as a XML message (which are XML-RPC calls), to read from it to which plugin and instruction set this call was sent, check in the scheme where this call is supposed to be delivered, and send it there. For delivering such functionality the core is using a HTTP Components library [13] created by Apache Software Foundation. This library delivers a complete toolset of low-level components focused on HTTP and associated protocols.

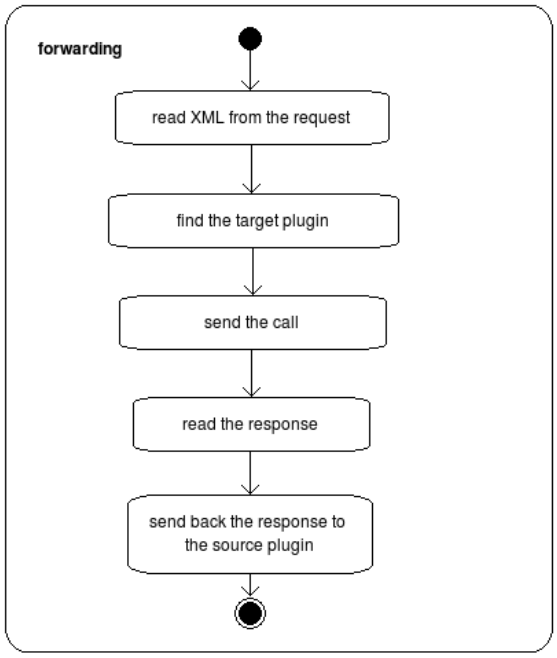

Fig. 12. Forwarding activity

The whole process of forwarding, illustrated in Fig. 12, creates for all plugins an illusion of dynamically changeable set of signals/ interaction methods. Additionally, it allows to dynamically change the scheme of connections, without disturbing work of plugins.

The final result is that the end-user can bind any application (which was prepared to work as an interface plugin) to any interaction plugins he wants, without worrying about compliance – if the plugin supports the same interaction set as the application – it will work smoothly.

7. OLIMPUS' possible applications

The OLIMPUS is not an application but a platform for building applications. Therefore OLIMPUS not providing any direct functionality for the end-users. However, it could be an invaluable, user friendly solution supporting using a computer just by adding the proper set of interaction and interface plugins.

The first area, which comes to mind, where OLIMPUS could be a useful and functional solution is an area of personal computing. Home users are greedy for custom human-computer interfaces, which makes using a computer more enjoyable. OLIMPUS would deliver to them an ability to adjust the way of interacting with their favourite applications without additional costs, or specialistic knowledge, just by adding new interaction plugins. The interaction plugins would be created by both individual creators, or companies.

Secondly, this solution could be a really helpful in creating dedicated sets of software. OLIMPUS would be a good basis for integrating software applications with created for them interaction methods. It would make such solutions easily customizable, with a perspective of future change of the way of communication with such application at low cost, or the future reuse of the interaction method in another OLIMPUS-based projects. Finally, as a well tested tool it is minimizing the risk of failures at system integration point.

The existence of such technology is widely needed, especially where the diversity of the interaction methods with the computer is the key factor. The perfect example is the market of software for disabled people [14]. There is often a need to adapt special software to specific needs of some special group of people. In most cases the level of their disability is quite diverse, so they need different ways of interaction with the computer. With OLIMPUS, one application can by used by many different patients, and the only things which are changing are interaction plugins, adapted to the level of disability of the user. OLIMPUS perfectly meets such dynamically changing requirements on demand and gives a possibility to reduce cost as well as time for adjusting a new configuration.

Once OLIMPUS is installed all that has to be done is to install and run new or lacking plugins which automatically will be attached to the whole working system. It means OLIMPUS could be used independently of needs of practically anyone. What is more, OLIMPUS could support any combination of plugins and interfaces including also those yet to be developed in the future. On the one hand, OLIMPUS offers extensible API for developers giving innumerable opportunities to create and design completely new plugins and interfaces limited only by their imagination. On the other hand, strictly defined API provides backward compatibility and ensures that all new plugins and interfaces will be working correctly even if it is hard to predict their functionality right now.

8. Summary

The development of non-standard ways of user-computer communicating is becoming more and more rapid. By now mainstream computers have enough power to bear interaction based on voice recognition or image analysis. Additionally, more and more projects are popping out, trying to deliver such interaction experience. But because of the lack of support and inability to utilize them with existing software, they remain unused. Maybe this situation can be changed by projects like OLIMPUS, which are trying to find an easy and cheap way of adapting such interaction mechanisms to the real world usage.

The key aspect is the ability to change the controller binding with the destination application without a need to make expensive changes. That ability is the very base of the OLIMPUS system. In the same way, the system can be adapted to the habits and different user preferences in the ways of controlling a computer system. Above all, OLIMPUS allows using different interaction methods to the same application, or using one interaction method to power in many applications. This kind of solution, focusing on the architectural side of the human-computer interaction problem, could be a proper approach to the problem of adapting custom interfaces into everyday use.

9. Bibliography

[1] E. Rafajłowicz, W. Rafajłowicz, A. Rusiecki, *Algorytmy przetwarzania obrazów i wstęp do pracy z biblioteką OpenCV*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2009.

[2] http://opencv.willowgarage.com/ – OpenCV library webpage.

[3] Intel Corporation, *Open Source Computer Vision Library, Reference Manual*, December 8, 2000.

[4] http://www.emgu.com/wiki/index.php/Main_Page – EmguCV wrapper webpage.

[5] J. Canny, *A Computational Approach to Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714 (1986).

[6] R.O. Duda, P.E. Hart, *Use of the Hough Transformation to Detect Lines and Curves in Pictures*, Comm. ACM, Vol. 15, pp. 11−15 (January 1972).

[7] http://www.naukawpolsce.pap.pl/palio/html.run?_Instance=cms_naukapl.pap.pl&_PageID=1&s=szablon.depesza&dz=stronaGlowna&dep=368265&data=&lang=PL&_CheckSum=-309941833 – press info about Cyber-Eye project.

[8] http://www.inference.phy.cam.ac.uk/opengazer/ – OpenGazer project webpage.

[9] S. Ferg, *Event-Driven Programming: Introduction, Tutorial, History*, Version 0.2, 2006-02-08.

[10] http://www.java.com/ – Java programming language webpage.

[11] http://www.xmlrpc.com/ – XML-RPC standard webpage.

[12] http://computer.howstuffworks.com/lan-switch.htm – article about LAN Switches.

[13] http://hc.apache.org/ – Apache HTTP Componenets library webpage.

[14] J. J. Cañas, "Technology for special needs", *An interdisciplinary Journal on Humans in ICT Environments*, ISSN: 1795-6889, April 2006.

[15] Z. Świerczyński, P. Rokita, „Podwyższanie rozdzielczości obrazów cyfrowych z wykorzystaniem informacji o krawędziach", *Biuletyn Instytutu Automatyki i Robotyki*, Nr 25, ITiA WAT, Warszawa, 2008.

Koncepcja architektury oprogramowania do komunikacji człowiek-maszyna

R. KASPRZYK, J. SROCZYŃSKI, K. STELMACH, P. SZADKOWSKI, K. WAWRUCH

W artykule przedstawiono koncepcje dwóch autorskich rozwiązań problemu komunikacji człowiek-maszyna. Obie koncepcje zostały zrealizowane w postaci prototypów o kodowych nazwach THEIA i OLIMPUS, potwierdzających słuszność założeń i ich realizowalność.

Projekt THEIA (*The Handy Eye-Interaction Adapter*) miał na celu zbudowanie platformy pozwalającej na swobodną interakcję z komputerem, za pomocą jedynie wzroku. Poprzez analizę obrazu ze standardowej kamery internetowej, program ustalał strefę ekranu, na którą patrzy aktualnie użytkownik. Wraz z nakładkami na system operacyjny, będącymi integralną częścią projektu THEIA, umożliwia to pełną, a przy tym prostą obsługę komputera.

OLIMPUS (*Overall, Light, Interactive Multi Purpose User-interface System*) jest platformą do tworzenia i integracji nowoczesnych interfejsów użytkownika z dowolnymi, niestandardowymi źródłami interakcji. Tym samym OLIMPUS upraszcza proces wytwarzania interfejsów symultanicznie współdziałających z wieloma źródłami interakcji i ich dynamiczną podmianę.

Słowa kluczowe: komunikacja człowiek-maszyna, interfejs człowiek-maszyna, architektura.

BIULETYN INSTYTUTU SYSTEMÓW INFORMATYCZNYCH 8 45−52 (2011)

Risk Management for the Needs of Critical Infrastructure

A. MACHNACZ
a.machnacz@policja.gov.pl

Police Headquarters
Domaniewska Str. 36/38, 02-672 Warsaw, Poland

The content of the article presents a proposal for implementing the risk management process for the protection of critical infrastructure as an important element in the process of ensuring the national security and its citizens, and the also the smooth functioning of public authorities as well as the public administration with its institutions and entrepreneurs. Various stages of risk management are discussed, including its planning, identification and analysis, and also risk response planning and its monitoring and control. A very important role was indicated in the process of risk management to ensure the security of critical infrastructure, including a presented proposal of risk calculation as a method of optimizing security costs.

Keywords: critical infrastructure, risk management, national security.

1. Introduction

The development of civilization and the rapid technological change marked the beginning of the fundamental transformation of the social structure and the way of life of the individual. As a result of technical progress television appeared, digital television, mobile phones, internet, personal computers, which revolutionized life. Increasing automation and the use of machines to perform operations previously carried out by people not only contributed to increased efficiency, but also to human dependence on this type of solutions. The changes enabled the easy and rapid exchange of information between offices, health centres, research institutes and universities, as well as between government institutions, and citizens and businesses. At the same time, it became possible and common to conduct daily activities through the internet, such as: online shopping, scheduling doctor appointments or managing official matters.

By observing the development of civilization it can be concluded that intensive technological change, and what comes after, the development of advanced information technology and automation has increased the quality and comfort, but also contributed to the growth of risks. Now, it would be difficult to function without well operating supply systems: energy, water and food, communications systems and networks, financial and transportation systems, or even rescue and chemical manufacturing systems. The violation of the continuous operation of the public administration might cause destabilization of a functioning of the nation. All these elements affect the security of the state and its citizens, including the effectiveness of authorities and public administrations as well as institutions and entrepreneurs, which are part of the so-called critical infrastructure [1], whose protection should be one of the main priorities of each country [2].

The development of technology puts the public authorities and other institutions under threats that could lead to a breach of the integrity and continuity of critical infrastructure. The development of mass media and the internet guaranteed flow of information that has become simultaneously a great tool in the hands of terrorist organizations allowing them to gain significant public and media effects by shaping the behaviour of the authorities and thus obtain the expected results in the destruction or disruption of critical infrastructure. Analogous consequences can result not only from intentional and unintentional human activities, but also may be the result of the forces of nature.

In order to counter threats of destruction, damage or disruption of the critical infrastructure, it becomes important to take coordinated actions by all entities in the area of protection of this infrastructure. One of these actions is the current performance of risk analysis and converting its results to the implementation of security measures. These actions are directly related to risk management, which consists of constant monitoring of the elements of this infrastructure and reducing the potential for adverse events by estimating values, the so-called risk factor relating, referring to the status of systems and including related operating facilities, equipment, installation and key services for the safety of the nation and its citizens [3], [7]. This overall process of identifying, controlling and eliminating or minimizing the likelihood of events that may affect critical infrastructure resources, requires a disciplined, consistent and reliable approach. This will ultimately allow avoiding the risk, taking preventive action or transferring the risk or its diversification. These activities are basic strategies to counter the risk, under which security measures are chosen.

Fig. 1. Stages of risk management
Source: own work

It is worth noting that risk management is a process consisting of well-defined, successive and mutually determining the stages, creating simultaneously a repeating cycle, as shown in fig. 1, in which the important element is the constant communication of any emerging risks. The effects of the actions taken and developed materials in one phase constitute a data source for the next step in the cycle of risk management. It is important that the selected processes, such as risk identification, take place continuously.

Later in the article, based on the PMBOK methodology [4] a proposition is presented of a risk management system for the protection of critical infrastructure allowing the current identification of risks, determining their size and identifying areas needing safeguards.

2. Planning Risk Management

Effective risk management, irrespective of the form and area of operation, first of all requires planning that consists of predicting how will various processes or events be developed in the future, as well as for programming and developing a plan.

Planning risk management for critical infrastructure protection should rely on the preparation and organization of the risk management process, under which will be determined, among other things: the thresholds of risk acceptance, the system of evaluation and interpretation of events, early warning indicators, and also the creation of the organizational structure. The task of such a structure should be to take action for the isolation, reduction and eventual elimination of these risks, as well as alternative methods of preparation, in order to protect against threats, which may occur during the planning and execution of works relating to the protection of critical infrastructure.

The result of the work undertaken within the framework of planning for the security of the critical infrastructure should be the creation of the so-called risk management plan containing [5]:

1. A general description of actions, as the basic presentation of all works, for which applied risk management will be used, including suggestions when and how risk management should be applied.
2. Description of the roles and responsibilities associated with them, as a list of duties and permissions of risk management for individual owners of elements of the critical infrastructure, including rules for reporting and issues escalation paths.
3. Description of the risk management process, as a representation of each of its steps, and an indication of when each of them will be used along with the eventual determination of any deviations in this area and the reasons for such deviations have been made, including subjecting the process characteristics to continuous tracking of risk.
4. Indicating categories, indicators, evaluation system and risk acceptance thresholds, including:
   - categories of risks and responses to risk depending on whether it is perceived as a threat or as an opportunity
   - indicators for early warning, i.e. risk materialization, by measuring changes in the critical areas regarding the functioning of critical infrastructure and to identify appropriate preventive actions as well as defining conditions and a time of their launch
   - ratings system, as a method of visualization of risks, which is used in risk analysis and for the interpretation of obtained results
   - thresholds for risk tolerance and acceptance thresholds, that is criteria for determining when actions should be taken in response to the identified risks.
5. Estimate the budget required to ensure efficient risk management.

6. Description of tools and techniques supporting the risk management process, including data sources that can be used.
7. Determining documentation requirements, including:
   - how to create documentation of the risk management process, including a description of the reporting system as a statement of purpose and destination, frequency, structure and required contents of reports
   - available document templates, their purpose and location, in order to achieve consistent reporting and the possibility of aggregation of information from individual reports, including the preparation of templates, including a risk response plan and risk register
   - principles of assuring control and quality of documents, their versions, how to store, uniform structure and format, mode of applying corrections, consistency between the documents as well as approval, review and feedback.
8. Description of the method of periodic review of the risk management plan as a way to verify the risk management practices, including checklists tailored to the type of activities and a list of issues requiring attention, including supplementary information sources.
9. The schedule of activities related to risk management.
10. Glossary of terms as an element allowing to unequivocally understand the various issues contained in the plan by all participating entities.

Included in the risk management plan the description of the actions and roles, as well as responsibilities, should present the main tasks of the critical infrastructure security area and the corresponding targets with specific timeframes, as well as identifying those entities responsible for their achievement. Moreover, the description should make demands on the process of circulation and approval of all documents created within the framework of risk management to prevent the omission of relevant entities, whose potential comments and suggestions could substantially affect the minimization of risk. Presenting the different steps of the risk management process used and specifying the conditions of their implementation should allow the visualization of the logical consistency of the entire approach to the management process for the sake of completeness and correctness of all operations. Bypassing any of the described steps could lead to omission of some factors that decide the target model of critical infrastructure security.

By considering the critical infrastructure certain categories of risk can be distinguished covering areas such as: technical, organizational, external and environmental. In each of these areas both risk and individual risk factors can be isolated. Such a division allows to organize the types of risks due to the source origin, and thus makes it possible to dedicate specialized resources for each category. The technical area may include all of these risks, which relate to the broader meaning of technology, including support automation solutions that are modern elements of critical infrastructure. The adopted structure of management and coordinating activities under the security of this infrastructure will form another category of risks. The group of external threats will be the factors arising from natural causes or from human activity, including terrorism, which may cause adverse changes in physical, chemical or biological resources, creations and constituents of the protected wildlife. To the last above-mentioned category, i.e. the group of environmental risks, can include risks arising from natural disasters, including: floods, fires and earthquakes.

To ensure the proper and orderly process of dealing with risks in the area of critical infrastructure security, the following response categories to risk can be specified [5]:

1. REDUCTION – take action to reduce the likelihood of the occurrence or minimize the effects, if the threat were to materialize.
2. ELIMINATION – changing the selected element associated with the operation or infrastructure security, such as: changing the management organization, technology, suppliers, production method, etc.
3. TRANSITION – transferring responsibility for specific risks to a third party, such as by purchasing insurance.
4. ACCEPTATION – a conscious decision to refrain from any combination of preventive and simultaneous constant monitoring of a given risk in terms of assessing whether they should still be tolerated.
5. SHARING – some form of risk sharing between parties according to established rules.

The risk management plan, cannot forget about the early warning indicators, the ratings system in terms of the interpretation of results and thresholds of tolerance and risk acceptance, which appropriate definition is the essence of all actions to secure critical infrastructure often crucial for an appropriate response to emerging events.

However, it is important to estimate the budget relating to risk management. Keep in mind that in practice often possessed financial resources affect the selection of protection mechanisms reducing identified risks, but also their absence or excessive limitation makes it impossible to implement all required safeguards. The risk management plan should also include a description of the specialized tools and proven techniques, which without their support would make it difficult to efficiently conduct, for example, a risk analysis or an appropriate risk response plan for such a huge list of resources that the critical infrastructure and threats associated with it creates. Examples of techniques, which can be used in various stages of risk management, as well as tools to perform procedures and fast calculations supporting these techniques may be classified as follows [5], [6]:

- TECHNIQUES: BPEST (Business, Political, Economic, Social, Technological) analysis, PESTLE (Political Economic Social Technical Legal Environmental), FMEA (Failure Mode & Effect Analysis) and CRAMM (CCTA Risk Analysis and Management Method)
- TECHNIQUES: Event tree analysis and fault tree analysis
- TECHNIQUES: Dependency modelling and real option modelling, and also decision-making in conditions of risk and uncertainty
- TOOLS: PEST Advanced Software Analysis Tool, SWOT Expert, CRAMM Expert, RISKAN, Deep SWOT analysis software, PILAR / EAR.

In addition to the above-mentioned techniques also used are questionnaires, prospecting, business studies and scenario analysis, industry benchmarking, risk assessment workshops or hazard & operability studies. The are many supporting software for these techniques, both commercial and open-source, such as SmartDraw or EBIOS.

Ensuring adequate quality of work requires the formulation of requirements for the created documents through the preparation of document templates. They allow repeated consistency assurance of the created documentation and to speed up work, especially if it is required, just as it is in the case of critical infrastructure, aggregating data from different entities. Described in the risk management plan, the monitoring and verification methods are to make the entire process practical and ensure its continuing topicality, consistency, completeness and consistency of requirements contained therein. The lack of control procedures and periodic review of the risk management plan could bring comparably irreversible effects, than the lack of the same plan. To ensure the preservation of the time regime of carried out tasks in various phases in the risk management plan the schedule of actions is defined showing the various stages and the expected results on the timeline maintaining the infinite cycle. Planning the course of activities in time helps to raise awareness of the scope as well as the relationship between them, it makes easier to supervise and the early detection of implementation threats. The richness of language causes that the placement of an adopted nomenclature into the risk management plan often allows for the exclusion of contradictions and confusion as to the importance of specific issues.

A high quality product in the form of a risk management plan translates directly into successive stages of management. It provides the mechanisms necessary to analyze and identify risk reduction measures, as well as it systematizes and organizes the approach to the entire process to ensure its completeness and consistency.

All work associated with creating and updating a risk management plan for critical infrastructure security, due to its nature and contents, should be subject to laws on the protection of classified information.

3. Risk Identification

Another action in the cycle of risk management is that its identification is reduced to detecting sources of risk and its description, and then systematizing risks according to accepted categories. Due to the nature of the steps the process should be performed several times during both the planning and in the endless cycle of risk management.

The description of risk is its term by giving its source, event (incident or accident), causing the risk and the reasons allowing the emergence of risk as well as what effects this risk calls. The event is the occurrence of or the change in specific circumstances that may occur once or repeatedly.

The process of identifying risks for the needs of critical infrastructure security should be an iterative process, carried out in all phases and stages of risk management. It should involve all entities, which are ruled by different infrastructure resources, including its complex nature and any functional dependence between its elements. It is appropriate to provide the information base and personally involve people with appropriate knowledge of the area.

Defined in an earlier phase of risk categories allow to qualify all unfavourable events that may occur during the planning and implementation of the continuous risk management process, and thus constitute a unified mechanism for collecting and processing such information.

The main result of the risk identification stage is a list of identified sources of risk in the area of critical infrastructure security. This list should also include a list of activities that trigger, i.e. symptoms and warning signals, indicating the circumstances of the occurrence of adverse events. The structure of the list of risk sources for the needs of critical infrastructure security could take the following form:

- risk identification, facilitating its monitoring and control
- category, cause and nature of the risks, which includes the given factor (risk, opportunity)
- the name or description of all or part of the critical infrastructure, which the risk concerns, including giving entity names in which the properties are specific elements of the infrastructure
- the effect of risk in the form of the description of the effect of the occurrence of a threat or opportunity
- risk symptoms as circumstances indicating that there has been or there will soon be a to risk occurrence.

It is important at this stage that no risks be ignored, whose sources are beyond the control of the authority responsible for critical infrastructure security. We should not neglect the study of individual results of side effects and identify possible causes and scenarios showing what effects may occur.

Implementation of the risk identification process is supported by techniques and tools identified in the planning stage of risk management. Predicting what events may significantly affect the non-fulfilment of planned tasks would be much harder without the possibility of using these techniques and specialized tools.

The proper execution of this stage is only possible if you have a good knowledge of the critical infrastructure, as well as an excellent knowledge of all related matters, including the environment in which it operates. Underestimating or omission of any of the risks would cover only a fragment of reality, which would affect the nature and meaning of risk management.

4. Qualitative and Quantitative Risk Analysis

Another element of the whole process of risk management is its analysis phase, during which the estimated size of the probability and consequences of the existence of previously identified risks. It is a process for understanding risk and to determine its level. Gathered here are the basis for risk assessment and decision-making in dealing with it. The results of this phase are the basis for further planning for adverse occurrences.

Carrying out risk analysis for the needs of critical infrastructure should begin in the so-called qualitative analysis consisting of putting identified risks in a hierarchy according to their potential impact on infrastructure security, including the possible distance in time. This operation should be based on the filtration of these risks, in order to determine, which of them can be accepted because of the minima probability of the occurrence and which and in what order should they be subject to further analysis and risk response plan. As a result of the qualitative analysis an approximate assessment of the risk factor probability and an estimate of risk significance or circumstances identified as a risk factor should be obtained. Repeating this type of analysis and testing obtained results will naturally allow the observation of trends that may constitute a legitimate basis for the decision to intensify or reduce operations involving risk management.

The result of the implementation of quality risk analysis for critical infrastructure security should be a list of risks in terms of priorities in planning and taking preventive actions as well as a list of risks for further analysis.

For the case of identifying risks of significant impact on critical infrastructure security the quality analysis should be supplemented with a quantitative analysis, involving the definition of measurable probability size values as well as the consequences of adverse events. Such a measurement can determine the chance of achieving an appropriate security infrastructure, and also determine the necessary reserve levels of time and cost, which may be needed to compensate for the effects of the occurrence of individual risks.

The group of basic tools and techniques used for quality risk analysis include scales and matrixes probability assessments as well as the impact of risk occurrence. For the quantitative analysis, tools are different from each other, because of the degree of complexity and they often include:

- survey research on the occurrence of certain events, in order to determine the probability size and the impact of the occurrence of risk
- simulation techniques for generating hypotheses concerning the probability of an occurrence of a particular scenario of operating conditions and the protection of critical infrastructure
- an analysis of decision trees, which allows to build sequence diagrams with their defined probability and costs, including each possible logical paths of events
- sensitivity analysis allows the determination, which risks have potentially the greatest impact on infrastructure security.

The result of quantitative risk analysis for critical infrastructure security should be:

- list of measured quantified risks in order of priority, which is used to plan and take preventive action
- Probabilistic analysis includes projections of cost and time of implementing tasks for infrastructure security, including the probability size of achieving the objectives of cost and time
- list of trends that characterize the results of quantitative risk analysis obtained on the basis of its execution several times and used for ranking risks in terms of priority in planning and taking preventive action.

Cumulative risk analysis may therefore be a combination of the above dependencies on the in relation to circumstances of the available information, data and resources. The mere presentation of the results of risk analysis can boil down to determining their impact and their probability on the timeline grouped by situation and expressed in a tangible or intangible method.

5. Planning Risk Reaction

Risk analysis has provided material support to make decisions about the priorities of conduct in relation to specific events. By taking definite steps takes into account the wider context of risk. This applies to the circumstances of the impact on authorities that do not bear direct responsibility or benefits with taking risks in relation to the authorities directly responsible for the risk and the outcome of actions resulting from decisions made. This can lead, in particular, to the decision of not proceeding with the risk in any way except for only using control measures on it.

Therefore, the key step in the risk management process is planning a response to risk allowing the indication of variations of proceeding in terms of eliminating the risks and increasing potential benefits. This process shows, based on the defined categories during the planning phase, the response to risk, possible reactions by indicating appropriate actions as well as assigning the responsible entities for carrying out actions related to the risks. Planning a response to the risk for critical infrastructure security should take into account such a plan of proceedings, so that actions taken would be most effective. Planned responses should be proportionate to the effects of adverse events, eliminate (or suppress) the impact of the given threat in a manner that is cost-effective and be implemented on time. This requires the involvement of many institutions, which are in charge of the individual elements of the infrastructure.

The main result of the reaction to risk planning phase for critical infrastructure security should be to prepare the so-called response to the risk plan with a list of residual and secondary risks, i.e. such that remain even after the implementation of strategies to reduce, eliminate, transfer, accept or even risk sharing. Risk response plan should take into account the results of qualitative and quantitative analysis of risks and the arrangements for their administrators, along with a description of actions that constitute a response to the risk. For each action a duration and financial resources should be specified, including other resources required to implement them. In case of risk acceptance contingency plans should be defined.

Planning response to risk is reduced to developing a plan for dealing with risk, which should include:

- choosing what kind of impact to have on risk
- selecting the mode of implementation of the given method, including the selection or modification of control measures
- prepare and implement a plan for dealing with risk.

Selection of how to impact on risk is a cyclical process and includes:

- conduct a risk assessment
- determine a tolerable level of risk

- assess the effectiveness of a given course of action.

Selecting the mode of implementation of a given course of action is reduced to the analysis of cost and effort necessary to implement the given mode in relation to the benefits. It is very important that each of the analyzed modes take into account legal requirements and other regulations including social responsibility and the impact on protecting the natural environment. The mere implementation of a given mode should be carried out according to the plan for dealing with risks, in which powers and responsibilities are clearly defined in relation to the given implementation.

The preparation and implementation of a plan for dealing with risk is aimed at documenting how to implement certain modes of conduct. They should include in particular:

- justifying the selection along with the expected benefits
- a list of people and responsibilities
- a list of indicators along with the limit values
- the way of communication and the rules for monitoring
- the required deadlines and schedule.

It is desirable that the plans of conduct be linked with the management processes of the given body responsible for the entire process.

6. Monitoring and Controlling Risk

Preparing adequate actions in the area of preventing critical infrastructure threats requires their implementation, current tracking of their execution and controlling the status of each identified risks.

Monitoring and controlling risk for critical infrastructure security requires maintaining an adequate reporting system by all entities, to whom competencies are specified for elements of infrastructure. This task comes down to constantly checking, whether the identification and evaluation of risk are properly done as well as if appropriate measures and solutions are applied, and also whether unidentified risks have already occurred. Observation of the identified risks, including residual and secondary risks, as well as implementing risk response plans and evaluating their effectiveness can determine whether preventive action taken in response to the risk yield expected results or require reconstruction. This should be a component of the risk management process. The very process of monitoring and control should be characterized by, among others:

- be subject to regular checks and supervision,
- be carried out periodically or ”ad hoc”
- have clearly defined responsibilities and scope
- provide new information for risk assessment
- enable the detection of trends in risk assessment
- document the results and enable reporting.

Components to monitor and control risk are:

- monitoring understood as a continual observation and checking of the status, in order to identify changes of the given parameter
- review understood as an action taken to determine the suitability and effectiveness of a given parameter
- audit understood as an independent and documented process of obtaining an objective assessment of the scope and effectiveness of a given parameter
- reporting understood as a form of communication used to inform entities the of the risk management process about the results of the given parameter or the entire process.

The result of the monitoring and control processes of risks for the needs of critical infrastructure security should be:

- assessment of the introduced measures in terms of their effectiveness
- list of necessary corrective actions necessary to take in case of ascertained errors during the implementation of the establishments resulting from the risk response plans
- proposal of actions related to newly identified risks.

The complexity of the critical infrastructure and the associated risks that are dynamic and operate in a changing environment, make it an important element in the infrastructure security process are the accepted principles of risk management. It is extremely important of carrying out periodic audits of their compliance with common acceptable standards, as well as the ability to introduce improvements.

7. Conclusion

In view of the still emerging threats to critical infrastructure the key issue is risk management. However, without a unified risk management system for such a broad spectrum it would be impossible to appropriately and effectively plan and implement security measures. Ignoring the risk management planning stage would have unimaginable consequences for the security of critical infrastructure, because the next steps in the identification, analysis and planning a response to risk would completely be deprived of mechanisms for a reliable identification of risks, their assessment and planning countermeasures.

Indication for the entities, who are authorities of the elements of critical infrastructure, preparing, for example, crisis management plans and critical infrastructure security plans of a homogenous risk management system, could contribute to a more effective process of identifying and assessing risks, thus optimizing the cost of building the critical infrastructure security.

An important element of risk management should not only protect critical infrastructure, but also develop such solutions, so that potential damage and disruption in its functioning would be probably short, easy to remove and would not cause additional losses for citizens and the economy. The essence of these tasks should not only come down to protecting critical infrastructure from threats, but also to reduce their effects and its rapid recovery in the event of failure, attacks and other events disrupting the proper functioning of the state and its citizens.

Due to the complexity of critical infrastructure, which proper functioning often depends on many entities and institutions of the public administration are essential formal arrangements with the scope of responsibilities that they take on themselves other participating sides in the security of critical infrastructure and responding to various threats, including the implementation of actions resulting from the contingency plans.

The demand of the necessity of documenting should be emphasized, which causes all activities related to risk management to be identified. All records also provide a basis for improvement (learning) of methods and tools, as well as the entire risk management process.

8. Bibliography

[1] Act of April 26, 2007 on Crisis Management (Journal of Laws 2007 No. 89 pos. 590 with amendments).

[2] A. Tyburska (red.), *Ochrona infrastruktury krytycznej*, WSPol Szczytno, 2010.

[3] ISO 31000:2009, Risk Management – Principles and Guidelines.

[4] *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)* – Fourth Edition, Project Management Institute (Corporate Author).

[5] *Management of Risk: Guidance for Practitioners*: 3rd Edition, M_o_R publication, OGC – Office of Government Commerce (Author), Great Britain, 2010.

[6] *A Risk Management Standard*, Published by AIRMIC, ALARM, The Institute of Risk Management – IRM, 2002.

[7] ISO/IEC 27005, Information technology – Security techniques – Information security risk management.

Zarządzanie ryzykiem na potrzeby infrastruktury krytycznej

A. MACHNACZ

W treści artykułu przedstawiono propozycję realizacji procesu zarządzania ryzykiem na potrzeby ochrony infrastruktury krytycznej jako istotnego elementu w procesie zapewnienia bezpieczeństwa państwa i jego obywateli, a także sprawnego funkcjonowania organów władzy i administracji publicznej oraz instytucji i przedsiębiorców. Omówiono poszczególne etapy zarządzania ryzykiem, w tym jego planowanie, identyfikację i analizę, a także planowanie reakcji na ryzyko i jego monitorowanie oraz kontrolowanie. Wskazano bardzo ważną rolę zarządzania ryzykiem w procesie zapewnienia bezpieczeństwa infrastruktury krytycznej, w tym przedstawiono propozycję kalkulacji ryzyka jako metody optymalizacji kosztów środków ochrony.

Słowa kluczowe: infrastruktura krytyczna, zarządzanie ryzykiem, bezpieczeństwo narodowe.

Analyzing the Possibility of Modeling the Specificity of an Individual Stock Market Investors' Behavior

P. MICHAŁOWSKI
pmichalowski@wat.edu.pl

Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw

The paper presents various methods of modeling an individual stock market investors' behavior. The author analyzed the possibility of using approaches related to the theory of expected utility, prospect theory, game theory and other studies on this subject. The article is the result of efforts to create an adequate model in order to investigate the specificity of the decision-making behavior of the so-called individual stock market investors.

Keywords: stock market investors, mass investors, models of behavior, cooperative many-player game, the game with a continuum of players, the expected utility.

1. Introduction

When in 1725 two first capital stock markets were created – Wall Street specialized in selling tobacco and slaves, and Broad Street specialized in selling eggs and butter, surely not many people expected how this capital market would grow and change over the years, decades and centuries.

A huge boom in the 20's created a situation where individual investors – non-institutional persons taking care of small transactions mainly – took a significant part in the stock markets' transactions.

The Polish stock market was created in Poland in 1989. It was created on April 12, 1991 by the State Treasury pursuant of the Foundation Act of Giełda Papierów Wartościowych (The Warsaw Stock Exchange). Four days later GPW held its first trading session. Individual investors made over 80% of the transactions in this market.

During the next twenty years more than a million Poles became interested in the stock market. The number of partnerships, sales and investors has been growing rapidly. Individual shareholder's share in the number of transactions has stabilized at the level of 50%. It is not a typical situation in the world. Other stock markets get a smaller share of individual shareholders than in Poland.

Due to expansion of the capital markets (including stock exchanges), many studies on how to model, describe and predict the behaviour of the market itself as well as its investors have appeared.

To be more specific, those studies cover such topics as: all kinds of behaviors, mechanisms of those behaviors, reactions on the behavior of others, as well as the whole spectrum of mutual interactions and individual decisions, which are being made on the stock markets every day.

This work is an overview of different opinions on this subject. Opinions, which are based on many methods, all having one goal – to understand the shareholders' behavior, because their way of action (individually or in a group) is least recognisable, least predictable and dependent on a great amount of factors.

2. Expected Utility Hypothesis

One of the hypotheses related to the investors' behavior in risky situations is the expected utility hypothesis. It was presented to a large group of people for the first time by John von Neumann and Oskar Morgenstern in their book *Theory of Games and Economic Behavior*. According to the book, individual shareholders' behaviour is based on the utility function U defined on a set of alternatives S at risk. The hypothesis assumes that the choice of a random event with the results in this set should proceed to get the maximum value of the utility function.

The starting point to the expected utility hypothesis is a utility theory, which concentrates on the choice from non random alternatives. The theory also assumes that every element x from the set S is assigned to a certain utility value $U(x)$.

The expected utility hypothesis goes one step further – it concentrates on modeling the consumer's choice when the consumer chooses between formalized lotteries, using random variables with the results belonging to the set of alternatives *S*, but when he does not choose between sure actions.

In order to market shareholders the elements of a set *S* for the example can describe the changes of several action courses, both individual and whole carts.

The expected utility hypothesis requires defining the expected utility function *V*(*p*). It is defined on the lotteries' set *P*, which are random variables. The variables' acceptable results belong to the set *S*. In this hypothesis there is one condition saying that, if *U*(*x*) is a utility function with area *S*, then every function *g*(*U*(*x*)), when g is an increasing function, they will lead to the same conclusions. The relationship between expected utility function *V*(*p*) and utility function *U*(*x*) is shown in the formula below:

$$V(p) = \int U(x) dF_p(x),$$

where $F_p(x)$ is a distribution function of random variable's *p* probability.

The axiomatic of the expected utility hypothesis requires from the decision maker's preference's relation to satisfy the following four axioms (*A*, *B*, and *C* are the lotteries):

- completeness axiom – says that the preference's relation is consistent, which means that the person is able to compare each of the two lotteries he/she may choose $A < B$ or $A > B$ or $A = B$
- axiom of transitivity – assumes that the preference's relation is transitive $A \geq B$ and $B \geq C \Rightarrow A \geq C$
- axiom of continuity – says that the preference's relation of the decision maker is continuous and there is a probability *p* that:
  $B = pA + (1-p)C$
- independence axiom – for every $t \in (0;1]$ and $A > B$:
  $tA + (1-t)C \geq tB + (1-t)C.$

Like every mathematical model, the expected utility hypothesis is an abstract and a simplified description of the reality. It does not guarantee that results will be acquired when using it are the ideal projection of human behaviour, but it surely delivers an additional knowledge, which gets the researchers closer to their final goals.

3. Perspective Theory

The perspective theory stands in opposition to the dominating in economical mainstream expected utility hypothesis. The perspective theory is a psychological theory created in 1979 by Daniel Kahneman and Amos Tversky. It explains the way of decision making in a risky condition. Creating this was preceded by experimental studies which, in authors' opinions, leads to the conclusion much more different than expected in utility hypothesis.

As a research method they selected a questionnaire, in which respondents had to choose one of the two options formulated as forecasts:

Which of the possibilities do you prefer: (6000,.001) or (3000,.002)?

where the first number is a winning prize amount and the second one is a probability of winning.

The results of the questionnaire helped the authors to jump to three conclusions (named effects):

- Certainty effect – preferred are those forecasts, which give the sure profit even if the alternative offers higher but unsure expected profit
- Rebound effect – forecasts of loss and profit are treated in a different way – in case when all the forecasts make losses, the risk is searched, and the reverse of certainty effect is shown
- Isolation effect – complicated problems are being simplified, the respondents concentrate on the differences between alternatives, and not on the similarities between them.

It can cause inconsistency of preference, because on the way of solving the problem affects how it has been formulated.

In the perspective theory assumptions – the theory that describes the decision making process, there is a belief that the process consists of two stages: processing and the evaluation stages.

In the first one we will concentrate on analysing the possible operations using such operation as: coding, simplifying, separating, and skipping. In other words: at the beginning we will choose the point against, which we rate the decisions' result, next we will simplify the forecasts with a complex structure and refuse the extreme probabilities (the smallest and the highest). The next step is separating undeniable objects from risky ones and finally removing all the common elements of considerable forecasts.

On the evaluation stage we classify available decisions based on the impact of probabilities of the decision's elements on the entire decision and subjective value of possible results of the decision to the reference point.

The evaluation function is defined by π(pi), and the weight function by *v*(xi).

The evaluation function, according to Tversky's and Kahneman's studies, is characterised by:

- alternatives are evaluated in terms of the reference point, which position is dependent on experiences and the current status, for example the person's wealth
- the evaluation function is concave for positive forecasts and convex for the negative ones
- the evaluation function is more flat for positive forecasts than for the negative ones.

The wealth function reflecting the influence of probability on the related decision, is characterised by:

- is increasing in interval [0;1] and gets values $\pi(0) = 0$ and $\pi(1) = 1$
- is sub additive for small probabilities, which means that rare incidents have bigger influence on decision making than it would result from their probabilities $\pi(rp) > r\pi(p)$ for $0 < r < 1$
- probabilities do not add each other to 1 $\pi(p) + \pi(1-p) <!$
- treats very rare incidents as impossible and very possible ones as sure.

As a result of operation at the processing stage and evaluation stages, every decision gets a value, which is a basic criteria of decision making.

4. Game Theory

When in 2001 (in Poland in 2002), the Ron Howard's film titled *Beautiful Mind* with Russell Crowe as the main actor, was played in cinemas, many people heard about game theory for the first time. Four Oscars, four Golden Globes, and dozens of other prizes. The winning film is a story about the life of John Forbes Nash Junior, co-laureate of the Nobel Prize in economy. The film has presented to the masses the field of mathematics that can be observed in everyday life of each one of us.

Game theory comes from the studies on gambling games. It focuses on analysing the optimal behavior in conflict situations. It is applicable in many areas of people's lives as well as the economy.

According to game theory, every conflict situation is a game, and every participant is a player. Every player chooses one behavior strategy, which leads him to the specific prize.

What varies game theory from decision theory is the fact that in game theory one player's decisions affect the other individuals' decisions.

Its application in economy is obvious and present in many fields like auctions, fusions, acquisitions, financial operations, and also stock market's investigations and operations.

Researches usually concentrate on one set of strategy. They are based on what is determined by rational standards. The most common strategy profile in game theory is the ”Nash equilibrium”, which is used in economics. It assumes that the end player strategy is optimum, when the choice of its opponents is set. In equilibrium no player has any reason to withdraw from his strategy – in this meaning it is stable.

When it comes to game types there are several criteria of choice:

- cooperative or non-cooperative games – in a non-cooperative game players make their decisions independently, in cooperative games decisions can be made by the group of players who can consult their decisions with each other
- symmetric and asymmetric games – in symmetric games the result depends only on the strategy employed, and not on who is using it. In asymmetric games different players have different sets of possible strategies
- constant-sum and non-constant-sum games – in a constant-sum game using any of the available strategies does not cause gain increase
- perfect information and imperfect information games – in perfect information games each player knows the decisions previously made by other players
- two-players and many-players games – categorisation of the number of players
- finite and infinite long games – in finite games each player's set of strategies is finished.

Obviously there are many other criteria of choice but for this paper those mentioned above are sufficient. It is also worthwhile to take under consideration other studies on this issue additionally specifying factors adopted for games in the free market.

As Owen proves [4] when trying to apply the theory of n-player games to economical analysis, it appears that small games (with not many players) are not suitable in free markets situations. For this purpose we need games with a higher player number – single player influence on other players gains is unnoticed. Those kinds of games are called games with a continuum of players.

Another problem is the fact that for *n*-players games it would always assume that the utility can be transferred through players. Instead of considering all possible results available for coalition *S* we can give a total amount, which can be gained and we can notice that the amount can be divided between the participants of this coalition in any way.

Considering the above criteria we can assume that stock market investments are games: uncooperative, asymmetric, non-constant-sum, imperfect information, and with continuum of players. And from this point other studies should start.

However, it is worthwhile to consider that the above acquisitions do not have to apply to a whole stock market. For example buying shares is significantly different in this approach to games in the future market. The capital intended for buying shares helps the company grow – in this case both buyer and a company could win.

5. Behavioral Economics and Behavioral Finance

Another field of science that tries to answer the questions about shareholder actions (but not only) is behavioral economics and related to it behavioural finances.

They have been transferred to the field of economic and financial analysis, using the researches about human, social, cognitive and emotional aspects in order to understand economic decisions made by consumers, debtors, and investors, as well as influence those decisions on market prices, funds deriving, and allocation of resources. The main area of interest are rationality boundaries (self-interest, self-control) of economic aspects. Created behavioural models are usually a mix of psychology and neoclassical economy theory.

Behavioral economics is a return to the roots, to the classical period when the economics and psychology were close to each other. The example of this is a work of Adam Smith *The Theory of Moral Sentiments* describing the psychological basics of individual behavior. But in the time when neoclassical economics has been formulated, many people tried to move away from psychology and create a natural science describing economical behavior in regards to economical aspects. A significant move to return to merge those two fields was the 1967 work of Gary Becker *Crime theory*. In later years similar influence have had the earlier mentioned books by Tversky and Kahneman.

In behavioral economics and finances we can list three main issues:

- heuristic – decisions are not always made based on rational analysis
- framing – when the problem is shown to a decision maker, then it influences on its action
- market inefficiency – there are explanations for observed market results, which are in contrast to rational expectations and market productivity.

Models in behavioral economics usually relate to observed market anomaly and modify the neoclassical approach, describing the decision makers as persons using heuristic and framing.

On the other hand, the very important question regarding behavioural finances is why do the shareholders make systematic mistakes and how do those mistakes affect the prices and returns creating market inefficiencies, for example under- and overreactions for information. Those are usually assigned to the shareholder's limited attention, excessive self--confidence, and self-imitation. Behavioral finances additionally pay attention on asymmetry between decision of acquisition and stopping resources and strong aversion to loss, especially when the loss has an emotional background.

Behavioral finances are being used in, for example, Thaler's model regarding the price reaction on information with two stages under reaction-adaptation-overreaction, which make up the price trend. It shows that overreaction is accompanied by the following phenomenon: average assets revenue after a series of good information is less than after a series of bad information. It comes to the situation when the market is too strongly reacting on the information. Opponents of behavioral finances and behavioral economics accuse them of being the anomalies set (finance) and have little reference to the actual market situation (economics). The group of fans of those theories is still very strong.

6. Other Approaches

Because the number of approaches and describing shareholders' behavior is big, it is impossible to describe all of them. But it is worthwhile to mention a few.

Tony Plummer in his book *Psychology of Technical Analysis* refuses classical method of analysis and assumes that it is impossible to explain the phenomenon of financial markets on the basis of the concept of the shareholder who is able to understand the entire economic phenomenon. In his publication he presented three reproaches:

- classical economy assumes that individuals behave in linear and mechanical ways
- it does not consider that individuals are under a strong influence of the community
- it does not consider mood changes.

Based on the results of the Gustav Le Bon's studies over crowd psychology, he claims that the shareholder is guided by the collective soul, which ”orders to think, act and feel differently than acted, thought and felt each individual”. He assumed that the community is not a summary or the average of its components, and processes made inside of it are much more complicated and untypical.

There is also another approach that concentrates on neurophysiologic determinations, in particular on the role of hormones that drive human behavior in excitement. Professor Andrew W. Lo with professor Thomas Lux have asked questions in regards to it:

- do reliable methods of quantity measurement of preferred risk grade exist?
- how do those preferences link to features of character and temperament?
- can we discover the biological and genetic factors which determine behavior of financial market players?

In emotional behaviour analysis they have presented two kinds of risk:

- Instrumental risk – oriented on reaching the exact target in the future. It assumes that the shareholder realises the need of taking the risk. In taking such a risk the major role is played by the conscious and reflective thinking process
- Stimulated risk – less controlled, taking this risk is a result of needing strong experiences, and is not focused on reaching the exact target.

Those assumptions confirm the research of Tomasz Zaleśkiewicz and Jacek Radomski about the individual Polish shareholder.

7. Conclusion

As can be seen and could be assumed there are many methods on how to model the behaviour of shareholders, especially individual ones and approaches to this problem. All of them have common features, for example a player's approach to risk. Some of the methods present behaviour of the shareholder as controlled only by the need of maximising the revenue and minimising the loss, some of the others consider the human, psychological, neurological factor that determines the investor's behavior not only in regards to the risk, but also to the situations he is currently in, to his emotional state, former experiences and many other elements, which can manipulate human rationality.

After acquiring this knowledge, it is probable that we cannot rely only on the forecast of loss and revenue – characteristic features should also be taken into consideration. On the other hand relying only on emotions could also be wrong, because the reference point would be the rational decision.

The most interesting thing at this stage would be the approach that game theory is a very wide and well researched but a still open field. In this case it would be interesting, if many types of games coexisted at the same time to form a full picture of operations and transactions of the stock market. Each of these games would have different realities, different player strategies and different results.

However, game theory itself would not be enough. The probability would be based on some other element characterising psychological aspects of every game dependent on the situation and prepared by wider studies of players' profiles determining their behavior in the face of different perspectives and possibilities. The natural choice here would be the perspective theory and/or expected utility hypothesis – however, it is worthwhile to get through some other researches at this time such as behavioral finances and economics.

A correct description of the behavior of mass shareholders requires a multicriteria approach based on several studies about profits maximising, approach to risk, psychological and neurological conditioning, player's experience in the following areas: knowledge in financial profits, actual knowledge and former transactions as well as influence on one player's decision on other individuals decisions.

8. Bibliography

[1] A. Ameljańczyk, „Rozwiązanie wielo-osobowej gry kooperacyjnej z C-jądrem mocy continuum", *Biuletyn WAT*, Nr 2, 1977.

[2] A. Ameljańczyk, „Rozwiązanie kompromi-sowe wieloosobowej gry kooperacyjnej względem punktu wyznaczającego z normy Rp (.) a rozwiązanie Schmeidlera", *Biuletyn WAT* , Nr 2, 1978.

[3] A. Ameljańczyk, *Teoria gier*, Wojskowa Akademia Techniczna, Warszawa, 1978.

[4] A. Ameljańczyk, Wektorowa optymalizacja w modelach decyzyjnych ze szczególnym uwzględnieniem modeli growych", *dodatek do Biuletynu WAT*, Nr 2, 1979.

[5] A.E. Roth, "Subsolutions and Supercove of Cooprative Games", *Mathematics of Operations Research*, Vol. 1, No. 1, 1976.

[6] A.K. Dixit, B.J. Nalebuff, *Sztuka strategii*, MT Biznes Sp. z o.o., Warszawa, 2009.

[7] D. Kahneman, A. Tversky, "Prospect Theory: An Analysis of Decision under Risk", *Econometrica*, XLVII, 263–291 (1979).

[8] D. Luce, H. Raiffa, *Gry i decyzje*, PWN, 1964.

[9] D. Schmeidler, "The nucleolus of a characteristic function game", SIAM *Journal of Applied Mathematics*, 17, 1969.

[10] E. Płonka, *Wykłady z teorii gier*, WPŚl., Gliwice, 2001.

[11] G. Owen, *Teoria gier*, PWN, Warszawa, 1975.

[12] J. Rosenmuller, *Game theory. Stochastics, Information, Strategies and Cooperation*, Kluwer Academic Publishers, Boston, 2000.

[13] J.v. Neumann, *Theory of Games and Economic Behavior*, Princeton University Press., Princeton, 1944.

[14] J. Watson, *Strategia – wprowadzenie do teorii gier,* WNT, 2005.

[15] K. Jajuga, T. Jajuga, *Inwestycje*, Wydawnictwo Naukowe PWN, Warszawa, 2002.

[16] M. Malawski, A. Wieczorek, H. Sosnowska, *Teoria gier*, PWN, 2004.

[17] M. Nowak, *Finanse behawioralne jako nowy kierunek analizy racjonalności inwestorów na rynkach finansowych*, Instytut Prawa i Administracji PWSZ w Sulechowie.

[18] P. Bezalel, S.Peter, *Introduction to the Theory of Cooperative Games*, Springer--Verlag Gmbh, 2007.

[19] R. Aumann, *Handbook of Game Theory with economic applications,* Vol. 1–3, Elsevier, Amsterdam, 2006.

[20] R. Aumann, S. Hart, *Handbook of Game Theory with Economie Applications*, Elsevier, Vol. 1, Vol. 2, Vol. 3, 2006.

[21] T. Plummer, *Psychologia rynków finansowych. U źródeł analizy technicznej*, WID Press, Warszawa, 1995.

[22] Wikipedia.org, Game Theory, http://en.wikipedia.org/wiki/Game_theory, wersja z dnia: 06.03.2011.

[23] Wikipedia.org, Giełda Papierów Wartościowych w Warszawie, http://pl.wikipedia.org/wiki/Gie%C5%82da_Papier%C3%B3w_Warto%C5%9Bciowych_w_Warszawie, wersja z dnia 05.03.2011.

[24] Wikipedia.org, Prospect Theory, http://en.wikipedia.org/wiki/Prospect_theory, wersja z dnia: 06.03.2011.

[25] Wikipedia.org, Expected Utility Hypotesis, http://en.wikipedia.org/wiki/Expected_utility_hypothesis, wersja z dnia: 06.03.2011.

[26] Wikipedia.org, Ekonomia behawioralna, http://pl.wikipedia.org/wiki/Ekonomia_behawioralna, wersja z dnia 06.03.2011.

Analiza możliwości modelowania specyfiki zachowań giełdowych inwestorów masowych

P. MICHAŁOWSKI

W artykule przedstawiono różne metody modelowania zachowań giełdowych inwestorów masowych. Dokonano analizy możliwości wykorzystania podejść związanych z teorią oczekiwanej użyteczności, teorią perspektyw, z teorią gier oraz inne badania dotyczące tego tematu. Artykuł jest wynikiem prac nad opracowaniem adekwatnego modelu pozwalającego badać specyfikę zachowań decyzyjnych tzw. giełdowych inwestorów masowych.

Słowa kluczowe: giełda, inwestorzy masowi, modele zachowań, wieloosobowa gra kooperacyjna, gra z continuum graczy, użyteczność oczekiwana.

Comparison Study of Test Case Allocation Schemes in Software Partition Testing

K. WORWA, J. STANIK
kazimierz.worwa@wat.edu.pl

Institute of Computer and Information Systems
Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw, Poland

The partition testing method is a commonly followed practice towards the selection of test cases. For partition testing, the program's input domain is divided into subsets, called subdomains, and one or more representatives from each subdomain are selected to test the program. The goal of such partitioning is to make the division of the program's input domain in such a way that when the tester selects test cases based on the subsets, the resulting test set is a good representation of the entire domain. The main aim of the paper is to analyse the fault-detecting ability of the partition testing method. Using effectiveness metrics for testing and partitioning schemes this paper makes a comparison of various test case allocation schemes in partition testing.

Keywords: software testing, partition testing, random testing.

1. Introduction

The software testing stage plays a critical role in the software development process, and consumes a significant amount of resources including time and money. To help raise the effectiveness and efficiency of testing activities many approaches have been conducted to find an optimal testing strategy. The testing strategy determines which test case set should be selected as a basis for the testing stage. The set of all relevant inputs to a program is usually referred to as the program's input domain. Exhaustive testing, which tests all possible inputs is generally impractical since the inputs domain is normally very large. Typically, testers can only afford to test a small part of the input domain. Random testing, also known as statistical testing, is input-domain-based and often a black-box method for software testing. Despite persisting criticism, random testing, which is presented and discussed by almost every book on software testing, continues to be viewed and advocated by some as a simple, practical, and often cost-effective method to assess software reliability.

Partition testing is one of the main techniques towards the selection of test data. For partition testing, the program's input domain is divided into subsets, called subdomains, and one or more representatives from each subdomain are selected to test the program. Path coverage is a typical example of partition testing. Random testing can be viewed as a degenerate form of partition testing in the sense that there is only one ”subdomain”, the entire program domain. Since there is no partitioning, random testing does not bear the overhead of partitioning and of keeping track of which subdomains have been tested or not. A comparison analysis of partition testing and random testing has attracted significant attention in the literature. Specific contributions have been made by Boland [1], Duran and Ntafos [2], Gutjahr [3], Hamlet and Taylor [4], Weyuker and Jeng [5], Chen and Yu [6, 7]. Duran and Ntafos [2], and Hamlet and Taylor [4], performed a series of simulations and experiments to compare the effectiveness of partition testing and random testing. Their results were at first surprising to many people. They found that even when partition testing was better than random testing at finding bugs, the difference in effectiveness was marginal. Thus, when the overhead of partitioning is relatively expensive to random testing, it is likely that random testing will be more cost effective than partition testing in terms of cost per fault found. Weyuker and Jeng [5] conducted a formal analysis of partition testing strategies and compared the effectiveness of partition testing and random testing. They found that partition testing can be an excellent testing strategy or a poor one, as compared to random testing. Essentially, they found that the effectiveness of a particular partition testing strategy depends on how well that strategy groups together the failure-causing inputs, that is, inputs which produce incorrect outputs. In particular, partition testing is most successful when the subdomain definitions are fault-based. Their findings were very helpful in explaining the seemingly counter-intuitive results in the empirical analysis of partition testing and random testing by Duran and Ntafos [2] as well as Hamlet and Taylor [4]. Gutjahr [3] has compared partition testing and random testing on the assumption that program failure rates are not known with certainty before testing and are, therefore, modeled by random variables. He has shown that under uncertainty, partition testing compares more favorably to random testing than suggested by prior investigations concerning the deterministic case.

Boland et. al. [1] have studied how the concepts of majorization and Schur-Convexity can be used in determining conditions under which partition testing is more effective than random testing. They have generalized some of the results of Weyuker and Jeng [5], Chen and Yu [6, 7] and Gutjahr [3] and established more precise conditions under which partition testing outperforms random testing.

Based on and inspired by the work of Weyuker and Jeng [5] and Chen and Yu [6, 7], this paper is aimed at adding some new results and generalizations.

2. Notation and Assumptions

Comparison analysis of partition and random testing will be conducted on the basis of notation and assumptions that have been used in papers of Weyuker and Jeng [5] and Chen and Yu [6].

For any general program *P*, its input domain will be denoted by ***D*** and the size of ***D*** is *d*, $d > 0$. Elements of ***D***, which produce incorrect outputs are known as failure-causing inputs, and we shall assume that there are *m* of them, $0 \le m \le d$. The remaining inputs $c = d - m$ will be called correct inputs. The failure rate θ, is defined as $\theta = m / d$. Assume also that the total number of test cases selected is *n*. When testing is done by dividing the program domain ***D*** into *k* subdomains, $k > 1$, these subdomains will be denoted by $\boldsymbol{D_i}$, where i=1, 2, ..., *k*. The *i*-th subdomain has a size of d_i and contains m_i failures causing inputs, $0 \le m_i \le d_i$, and $c_i = d_i - m_i$ correct inputs, and has failure rate of $\theta_i = m_i / d_i$. The number of test cases selected from the *i*-th subdomain is n_i, $n_i \ge 1$. Of course, we have

$$\sum_{i=1}^{k} n_i = n . \tag{1}$$

Since in general testing performance increases with the number of test cases, we assume that partition testing and random testing have the same number of test cases when compared. We assume that all subdomains are disjoint. Therefore,

$$\sum_{i=1}^{k} d_i = d\ ,\quad \sum_{i=1}^{k} m_i = m\ ,\quad \sum_{i=1}^{k} c_i = c\ . \tag{2}$$

All random selections are also assumed to be independent and based on a uniform distribution. This means that when a test case is selected from ***D*** or $\boldsymbol{D_i}$, the probability that it is a failure-causing input will be exactly θ or θ_i respectively. We are interested in the probability that at least one failure-causing input is found when all test cases have been selected and executed. This will be denoted by P_r for random testing, and P_p for partition testing. They are defined by the following formulas [5]:

$$P_r(n) = 1 - (1-\theta)^n \tag{3}$$

and

$$P_p(\mathbf{n}) = 1 - \prod_{i=1}^{k} (1-\theta_i)^{n_i}\ . \tag{4}$$

where $\mathbf{n} = (n_1, n_2, \ldots, n_k)$, $k > 1$, is a vector of the number of test cases n_i selected from the *i*-th subdomain.

If there is only one subdomain, i.e. $k = 1$, we have $P_p = P_r$. Similarly, if $\theta_1 = \theta_2 = \ldots = \theta_k = \theta$ we have

$$P_p(\mathbf{n}) = 1 - \prod_{i=1}^{k} (1-\theta_i)^{n_i} = 1 - \prod_{i=1}^{k} (1-\theta)^{n_i} =$$
$$= 1 - (1-\theta)^{n_1 + n_2 + \cdots + n_k} = 1 - (1-\theta)^n =$$
$$= P_r(n).$$

i.e. above probabilities P_r and P_p are the same.

3. Comparison of Partition and Random Testing

Intuitively speaking, partition testing should be more effective in revealing program errors than random testing. However, Duran and Ntafos [2] as well as Hamlet and Taylor [4] observed in their simulation investigations that there is only a marginal difference in effectiveness between these two methods. Random testing might even be more effective than partition testing, especially when partitioning and associated costs of partition testing are high. We note, that intuitively, partition testing should be more effective, if more test cases are selected from subdomains having higher failure rates. However, this is insufficient to guarantee that it will be better than random testing. This is illustrated in example 1.

Example 1.

Case	d_i		m_i		n_i		θ_i		θ	P_r	P_p
	$k = 3$										
1	d_1	1200	m_1	10	n_1	10	θ_1	0.0083	0.01	0.1821	0.1912
	d_2	500	m_2	7	n_2	7	θ_2	0.014			
	d_3	300	m_3	3	n_3	3	θ_3	0.01			
Total		2000		20		20					
2	d_1	1200	m_1	11	n_1	13	θ_1	0.0092	0.01	0.1821	0.1725
	d_2	500	m_2	7	n_2	3	θ_2	0.014			
	d_3	300	m_3	2	n_3	4	θ_3	0.0067			
Total		2000		20		20					
3	d_1	1200	m_1	12	n_1	13	θ_1	0.01	0.01	0.1821	0.1821
	d_2	500	m_2	5	n_2	3	θ_2	0.01			
	d_3	300	m_3	3	n_3	4	θ_3	0.01			
Total		2000		20		20					

In this example, we have three partitions and both the total number of test cases that produce incorrect outputs and the total number of test cases are kept the same. In the first case partition testing is more effective than random testing, in the second case it is inversely, and in the third case probabilities P_r and P_p are the same.

In general, we do not have a priori precise information about the distribution of the failure-causing inputs so as to assess the performance of partition testing. Despite the absence of information about the distribution of failure-causing inputs, Weyuker and Jeng [5] did find a way of partitioning the program domain that is guaranteed to be not worse than random testing. They have proved that, if $d_1 = d_2 = \cdots = d_k$ and $n_1 = n_2 = \cdots = n_k$, then $P_p \geq P_r$. If, in addition, the failure-causing inputs are equally divided among the subdomains, so that $m_1 = m_2 = \cdots = m_k$, then $P_p = P_r$. This effect is illustrated in example 2. In other words, we will never do worse than random testing by having equal-sized subdomains and equal number of test data.

This result gives a sufficient condition of partition testing to be better than random testing. However, when we divide the program input domain according to the commonly accepted criteria, it is extremely unusual for the subdivisions to be of equal size. An obvious, but not intuitively appealing criterion is equal size partition**,** that is dividing the program input domain into subdomains of equal sizes.

In example 2 we have two cases with three partitions. In the first case $d_1 = d_2 = d_3$ and $n_1 = n_2 = n_3$, and we can see that $P_p \geq P_r$. In the other case additionally there is $m_1 = m_2 = m_3$ and probabilities P_r and P_p are the same.

Example 2.

Case	d_i		m_i		n_i		θ_i		θ	P_r	P_p
	$k = 3$										
1	d_1	700	m_1	10	n_1	50	θ_1	0.0143	0.0100	0.7785	0.7788
	d_2	700	m_2	7	n_2	50	θ_2	0.0100			
	d_3	700	m_3	4	n_3	50	θ_3	0.0057			
Total		2100		21		150					
2	d_1	700	m_1	7	n_1	50	θ_1	0.0100	0.0100	0.7785	0.7785
	d_2	700	m_2	7	n_2	50	θ_2	0.0100			
	d_3	700	m_3	7	n_3	50	θ_3	0.0100			
Total		2100		21		150					

Chen and Yu [6, 7] have generalized that result, that if the subdomains are disjoint, then partition testing is better than random testing so long as test cases are selected in proportion to the size of subdomains. Formally, that means that if $k \geq 2$ and $n_1/d_1 = n_2/d_2 = \cdots = n_k/d_k$, then $P_p \geq P_r$. This effect illustrates example 3 for $k = 3$ and $n_1/d_1 = n_2/d_2 = n_1/d_1 = 0{,}06$.

Example 3.

	k = 3											
Case	**d_i**		**m_i**		**n_i**		**θ_i**		**θ**	**P_r**	**P_p**
1	d_1	500	m_1	10	n_1	30	θ_1	0.0200	0.0095	0.6996	0.7019
	d_2	300	m_2	7	n_2	18	θ_2	0.0233			
	d_3	1300	m_3	3	n_3	78	θ_3	0.0023			
Total		2100		20		126					
2	d_1	500	m_1	10	n_1	30	θ_1	0.0200	0.0143	0.8371	0.8377
	d_2	300	m_2	10	n_2	18	θ_2	0.0333			
	d_3	1300	m_3	10	n_3	78	θ_3	0.0077			
Total		2100		30		126					
3	d_1	500	m_1	20	n_1	30	θ_1	0.0400	0.0286	0.9742	0.9747
	d_2	300	m_2	20	n_2	18	θ_2	0.0667			
	d_3	1300	m_3	20	n_3	78	θ_3	0.0154			
Total		2100		60		126					

The total number of test cases $n = 2100$ is kept the same for every case of three cases, but the total number of test cases that produce incorrect outputs $m = m_1 + m_2 + m_3$ is different for every case. We can notice that for every case in example 3 partition testing is more effective than random testing. Also, we can see that the increase of the total number of test cases that produce incorrect outputs $m = m_1 + m_2 + m_3$ entails the increase of both probabilities P_r and P_p.

It is noteworthy that the values of P_p in example 3 are only a little bit better than values of P_r. We will explain this fact in later parts of this paper.

4. The Best and the Worst Cases for Partition Testing

In this section, we analyze the best and worst cases of subdomain testing, as well as the conditions under which subdomain testing performs better or worse than random testing.
For the reasons of assessment of the schemes of program partitioning that have been proposed in papers of Weyuker and Jeng [5] and Chen and Yu [6] we will determine conditions to maximize the value of probability P_p.

Before we present the main result of this paper, we need to prove the following lemma first as it is needed for the proof of the theorem determining conditions to maximize and minimize the value of probability P_p.

Lemma 1. Suppose that $k = 2$, $n_1 \geq 1$, $n_2 \geq 1$ and the subdomains are renumbered in such a way that $\theta_1 \leq \theta_2$. Then:

$$(1-\theta_1)^{n_1}(1-\theta_2)^{n_2} \geq (1-\theta_1)(1-\theta_2)^{n_1+n_2-1}. \quad (5)$$

Proof. Finding the logarithm of the last inequality we have:

$$\ln[(1-\theta_1)^{n_1}(1-\theta_2)^{n_2}] \geq \ln[(1-\theta_1)(1-\theta_2)^{n_1+n_2-1}]$$

By using fundamental qualities of the logarithm function we obtain:

$$n_1 \ln(1-\theta_1) + n_2 \ln(1-\theta_2) \geq$$
$$\geq \ln(1-\theta_1) + (n_1 + n_2 - 1)\ln(1-\theta_2).$$

Then, after transformation we have:

$$(n_1 - 1)\ln(1-\theta_1) \geq (n_1 - 1)\ln(1-\theta_2).$$

It is obvious that if $n_1 - 1 = 0$ then above inequality is true.
If $n_1 - 1 > 0$, we obtain $\ln(1-\theta_1) \geq ln(1-\theta_2)$, and finally $(1-\theta_1) \geq (1-\theta_2)$.
The last inequality is true because we have assumed that $\theta_1 \leq \theta_2$.

From lemma 1 follows immediately that if $k = 2$, $n_1 + n_2 = n$ and $\theta_1 \leq \theta_2$, we have:

$$P_p(\mathbf{n}) = 1-(1-\theta_1)^{n_1}(1-\theta_2)^{n-n_1} \le \\ \le 1-(1-\theta_1)(1-\theta_2)^{n-1}.$$

Making use of the lemma 1 we can prove the following theorem.

Theorem 1. Suppose that $k>1$, $\sum_{i=1}^{k} n_i = n$ and the subdomains are renumbered in such a way that $\theta_1 \le \theta_2 \le \ldots \le \theta_k$. Then we have:

$$\prod_{i=1}^{k}(1-\theta_i)^{n_i} \ge \prod_{i=1}^{k-1}(1-\theta_i)(1-\theta_k)^{n-k+1}. \quad (6)$$

Proof. Left side of the last inequality can be written as follows:

$$\prod_{i=1}^{k}(1-\theta_i)^{n_i} = (1-\theta_1)^{n_1}(1-\theta_2)^{n_2}\prod_{i=3}^{k}(1-\theta_i)^{n_i}.$$

Based on lemma 1 we have:

$$\prod_{i=1}^{k}(1-\theta_i)^{n_i} = (1-\theta_1)^{n_1}(1-\theta_2)^{n_2}\prod_{i=3}^{k}(1-\theta_i)^{n_i} \ge \\ \ge (1-\theta_1)(1-\theta_2)^{n_1+n_2-1}\prod_{i=3}^{k}(1-\theta_i)^{n_i} \ge$$

$$\ge (1-\theta_1)(1-\theta_2)(1-\theta_3)^{n_1+n_2+n_3-2}\prod_{i=4}^{k}(1-\theta_i)^{n_i} \ge \cdots \\ \ge \prod_{i=1}^{k-1}(1-\theta_i)(1-\theta_k)^{n-k+1}.$$

The following corollary is the immediate result of theorem 1.

Corollary 1. If $k>1$, $\sum_{i=1}^{k} n_i = n$ and the subdomains are renumbered in such a way that $\theta_1 \le \theta_2 \le \ldots \le \theta_k$, then:

$$P_p(\mathbf{n}) = 1-\prod_{i=1}^{k}(1-\theta_i)^{n_i} \le \\ \le 1-\prod_{i=1}^{k-1}(1-\theta_i)(1-\theta_k)^{n-k+1}. \quad (7)$$

The above corollary from theorem 1 can be present in following form:

$$P_p(1,1,\ldots 1,n-k+1) = \max_{\mathbf{n}} P_p(\mathbf{n}), \quad (8)$$

where $\mathbf{n} = (n_1, n_2, \ldots, n_k)$.

It is obvious that the following theorem is also true.

Theorem 2. Suppose that $k>1$, $\sum_{i=1}^{k} n_i = n$ and the subdomains are renumbered in such a way that $\theta_1 \ge \theta_2 \ge \ldots \ge \theta_k$. Then we have:

$$\prod_{i=1}^{k}(1-\theta_i)^{n_i} \le \prod_{i=1}^{k-1}(1-\theta_i)(1-\theta_k)^{n-k+1}. \quad (9)$$

The proof of theorem 2 is very similar to the proof of theorem 1 and will be omitted.

Using theorem 2 we can formulate the following corollary:

Corollary 2. If $k>1$, $\sum_{i=1}^{k} n_i = n$ and the subdomains are renumbered in such a way that $\theta_1 \ge \theta_2 \ge \ldots \ge \theta_k$, then

$$P_p(\mathbf{n}) = 1-\prod_{i=1}^{k}(1-\theta_i)^{n_i} \ge \\ \ge 1-\prod_{i=1}^{k-1}(1-\theta_i)(1-\theta_k)^{n-k+1}. \quad (10)$$

Thus

$$P_p(1,1,\ldots 1,n-k+1) = \min_{\mathbf{n}} P_p(\mathbf{n}), \quad (11)$$

where $\mathbf{n} = (n_1, n_2, \ldots, n_k)$.

The following two examples 4 and 5 illustrate correctness of corollaries 1 and 2.

Example 4 describes the best and the worst case of probability P_p for $k = 3$, $n = 30$ and $\theta_1 \le \theta_2 \le \theta_3$. In accordance with theorems 1 and 2 the probability P_p has the maximum for $\boldsymbol{n} = (1, 1, 28)$ and the minimum for $\boldsymbol{n} = (28, 1, 1)$ respectively. It should be noted that for $\boldsymbol{n} = (10, 15, 5)$ we have $n_1/d_1 = n_2/d_2 = n_3/d_3 = 0{,}025$, so that is a partition compatible with condition found by Chen and Yu [6]. The row that fulfills the Chen and Yu condition is shaded. We have previously stated that this condition guarantees that $P_p \ge P_r$, but the value of P_p is only a little bit better than the value of P_r.

Example 4.

n	***d***	***m***	***θ***
30	1200	60	0.05

m_1	10
m_2	20
m_3	30

d_1	400
d_2	600
d_3	200

θ_1	0.025
θ_2	0.033
θ_3	0.150

n_1	n_2	n_3	P_p	P_r
1	1	28	0.9900	
1	2	27	0.9887	
1	3	26	0.9871	
1	4	25	0.9854	
1	5	24	0.9833	
...				
10	12	8	0.8592	
10	13	7	0.8398	
10	14	6	0.8178	
10	15	5	0.7928	0.7854
10	16	4	0.7644	
10	17	3	0.7321	
10	18	2	0.6953	
...				
26	2	2	0.6504	
26	3	1	0.6025	
27	1	2	0.6474	
27	2	1	0.5990	
28	1	1	0.5956	

In this example we have: $P_p(1,1,28) = 0.9900$, $P_p(10,15,5) = 0.7928$ and $P_p(28,1,1) = 0.5956$. Thus in fact, the probability P_p is much greater than the probability P_r (30), i.e. $P_p(10,15,5) = 0.7928 > P_r(30) = 0.7854$ but simultaneously $P_p(10,15,5) = 0.7928$ is significantly worse than $P_p(1,1,28) = 0.9900$.

Example 5 describes the best and the worst case of probability P_p for $k = 2$, $\theta_1 \le \theta_2$ and $n = n_1 + n_2 = 100$.

In accordance with theorems 1 and 2 the probability P_p has the maximum for $\boldsymbol{n}$ = (1, 99) and the minimum for $\boldsymbol{n}$ = (99, 1) respectively. It should be noted that for $\boldsymbol{n}$ = (25, 75) we have $n_1/d_1 = n_2/d_2 = 0{,}005$. The row that fulfills that condition is shaded.

Based on example 5 we can formulate the following observation: if we have only two partitions (k = 2) and $\theta_1 \le \theta_2$, than:

1) $P_p \ge P_r$, if and only if $n_1/d_1 < n_2/d_2$ (above and with shaded row);
2) $P_p < P_r$, if and only if $n_1/d_1 > n_2/d_2$ (below shaded row).

If $\theta_1 \ge \theta_2$, than the above relations are inverse.

Example 5.

k	2
m_1	80
m_2	320
d_1	5000
d_2	15000

θ_1	0.0160
θ_2	0.0213

n_1	n_2	P_p	P_r	n_1 / d_1	n_2 / d_2
1	99	0.8836		0.0002	0.0066
2	98	0.8830		0.0004	0.0065
3	97	0.8824		0.0006	0.0065
4	96	0.8817		0.0008	0.0064
5	95	0.8811		0.0010	0.0063
	…			…	
21	79	0.8703		0.0042	0.0053
22	78	0.8696		0.0044	0.0052
23	77	0.8689		0.0046	0.0051
24	76	0.8681		0.0048	0.0051
25	75	0.8674	0.8674	0.0050	0.0050
26	74	0.8667		0.0052	0.0049
27	73	0.8660		0.0054	0.0049
28	72	0.8652		0.0056	0.0048
29	71	0.8645		0.0058	0.0047
30	70	0.8638		0.0060	0.0047
	…			…	
31	69	0.8630		0.0062	0.0046
96	4	0.8050		0.0192	0.0003
97	3	0.8039		0.0194	0.0002
98	2	0.8029		0.0196	0.0001
99	1	0.8018		0.0198	0.0001

5. Conclusions

In this paper, we have extended and generalized some of the results by Weyuker and Jeng [5] and Chen and Yu [6]. We have given more general characterizations of the worst case for partition testing, along with a precise characterization of when this worst case is as good as random testing.

Almost all program testing strategies share a common characteristic: the program's input domain is divided into subsets, called subdomains, and one or more representatives from each subdomain are selected to test the program. This approach to the selection of test data is commonly referred to as partition testing. In contrast to the systematic approach of partition testing, random testing simply requires test cases to be randomly selected from the entire input domain. Most of the related work on partition testing deals only with the assumption that all subdomains are disjoint. Duran and Ntafos [2] in their simulation and empirical work on the comparison of partition and random testing concluded that the two methods are almost equally effective, even under assumptions that seem to favor partition testing. Their results also indicate that random testing may often be more cost effective than partition testing. We have analyzed some of the results by Weyuker and Jeng [5] and Chen and Yu [6, 7] concerning conditions for partition testing to be better than random testing. Weyuker and Jeng [5] have found that if $d_1 = d_2 = \cdots = d_k$ and $n_1 = n_2 = \cdots = n_k$, then the partition method would not be worse than random testing. Chen and Yu [6] have extended and generalized some of the results by Weyuker and Jeng [5]. They have given more general characterizations of the worst case for partition testing, along with a precise characterization of when this worst case is as good as random testing. The most important result of their study is that as long as we select the number of test cases proportional to the size of the subdomains, partition testing is guaranteed to be no worse than random testing. In practice, most partition testing strategies divide the program domain into unequal-sized subdomains. If the sizes of the subdomains are known, then we know how the test cases should be distributed, so that the partition method would not be worse than random testing.
In many cases, however, the exact sizes of the subdomains may not be easily known. In that situation it is sufficient to know the ratio of the subdomain sizes in order to be able to determine the distribution of test cases.

In this paper, we have determined conditions to both maximize and minimize the value of probability P_p, in the sense of allocation of test cases to the subdomains.

The most important result of this paper is to find out that according to conditions for partition testing is guaranteed to be no worse than random testing, that have been determined by Weyuker and Jeng [5] and Chen and Yu [6], the difference of values of P_p and P_r is marginal. It was illustrated in Example 4, where for $k = 3$, $n = 30$, $d = 1200$ and $m = 60$ we have $P_r = 0.7854$, whereas for $m_1 = 10$, $m_2 = 20$, $m_1 = 10$, $m_3 = 30$ and $d_1 = 400$, $d_2 = 600$, $d_3 = 200$ and $n_1 = 10$, $n_2 = 15$, $n_1 = 5$ we have fulfillment of condition by Chen and Yu [6] $(n_1 / d_1 = n_2 / d_2 = n_3 / d_3 = 0.025)$ and $P_p = 0.7928$. Thus, $P_p - P_r = 0.0074$, while $P_p^{\max} - P_r = 0.2046$.

6. Bibliography

[1] P.J. Boland & H. Singh & B. Cukic, ”Comparing partition and random testing via majorization and Schur functions”, *IEEE Transactions on Software Engineering*, Vol. 29, pp. 88–94, January 2003.

[2] J.W. Duran & S.C. Ntafos, ”An evaluation of random testing”, *IEEE Transactions on Software Engineering*, Vol. SE-10, pp. 438–444, July 1984.

[3] W.J. Gutjahr, ”Partition testing vs. random testing: the influence of uncertainty”, *IEEE Transactions on Software Engineering*, Vol. 25, pp. 661–674, September/October 1999.

[4] R. Hamlet & R. Taylor, ”Partition testing does not inspire confidence”, *IEEE Transactions on Software Engineering*, Vol. SE-16, pp. 1402–1411, December 1990.

[5] E.J. Weyuker & B. Jeng, ”Analyzing partition testing strategies”, *IEEE Transactions on Software Engineering*, Vol. SE-17, pp. 703–711, July 1991.

[6] T.Y. Chen & Y.T. Yu, ”On the relationship between partition and random testing”, *IEEE Transactions on Software Engineering*, Vol. SE-20, pp. 977–980, December 1994.

[7] T.Y. Chen & Y.T. Yu, ”A more general sufficient condition for partition testing to be better than random testing”, *Information Processing Letters*, Vol. 57, pp. 145–149, 1996.

Porównanie wybranych strategii losowego testowania oprogramowania

K. WORWA, J. STANIK

W artykule przedstawione są wyniki porównania dwóch, najczęściej wykorzystywanych w praktyce, strategii losowego tworzenia zbioru danych testowych. Pierwsza z tych strategii, nazywana testowaniem w pełni losowym, polega na losowaniu poszczególnych przypadków testowych ze zbioru wszystkich możliwych zestawów danych wejściowych rozpatrywanego programu, przy czym najczęściej przyjmuje się tutaj, że wylosowanie każdego z tych zestawów jest jednakowo prawdopodobne. Druga z analizowanych strategii zakłada podział całego zbioru wszystkich możliwych zestawów danych wejściowych programu na tzw. partycje, będące podzbiorami, tworzonymi w oparciu o kryteria wykorzystywane w testowaniu strukturalnym. Strategia ta jest nazywana strukturalnym testowaniem losowym. Zawarte w artykule rozważania mają na celu określenie warunków, dla których jedna z ww. strategii testowania losowego jest lepsza od drugiej, w sensie prawdopodobieństwa wykrycia co najmniej jednego błędu.

Słowa kluczowe: testowanie oprogramowania, strukturalne testowanie losowe, testowanie w pełni losowe.